HD 63701 X0P (A,B)

8-Bit CMOS Microcomputer (ZTAT)

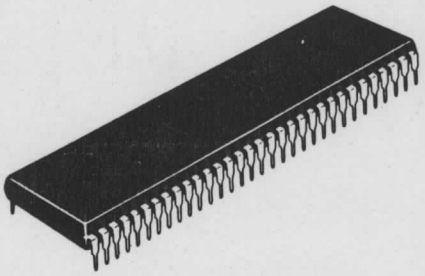
HITACHI

The HD63701X0 is a high performance 8-bit CMOS single chip microcomputer unit (MCU) which, including 4k bytes of PROM, is pin compatible with the HD6301X0.

The HD63701X0 contains 4k bytes of PROM, 192 bytes of RAM, serial communication interface and 53 parallel I/O pins in addition to CPU. It includes functions of halt, memory ready, low speed access and releasing external bus at system expansion.

The HD63701X0 is available in a 64-pin shrunk plastic package. It can be programmed in the same procedure as 2732A type EPROM.

HD63701X0P, HD637A01X0P,
HD637B01X0P

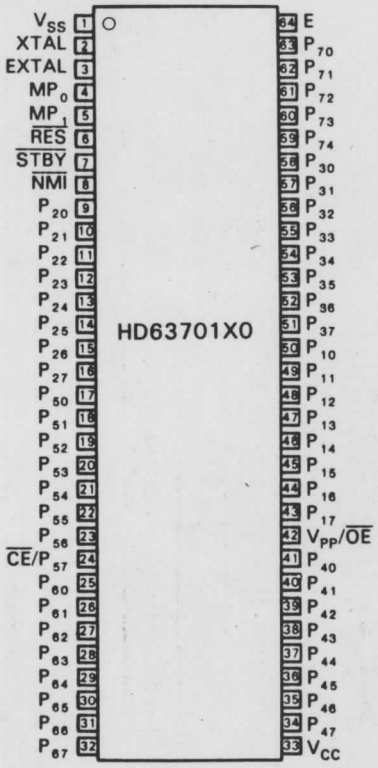


(DP-64S)

■ FEATURES

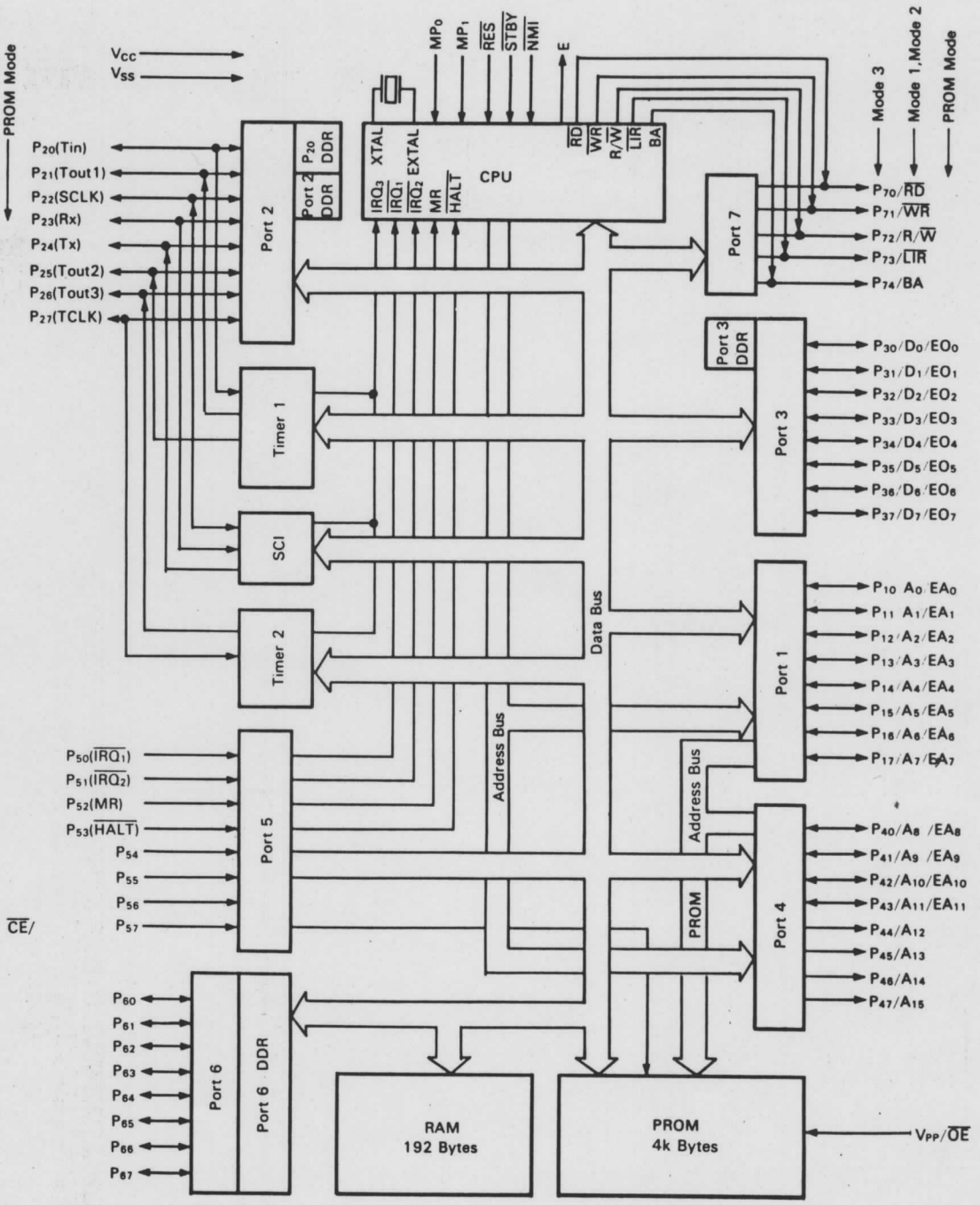
- Instruction Set Compatible with the HD6301X0
- 4k Bytes of PROM (compatible with 2732A type)
- 192 Bytes of RAM
- 53 Parallel I/O Pins
  - 24 I/O Common Pins (Port 2, 3, 6)
  - 21 Output Pins (Port1, 4, 7)
  - 8 Input Pins (Port 5)
- Driving Darlington Transistor (Port 2, 6)
- 16-bit Programmable Timer
  - Input Capture Register x 1
  - Free Running Counter x 1
  - Output Compare Register x 2
- 8-bit Reloadable Timer
  - External Event Count
  - Spare Wave Occurrence
- Serial Communication Interface (SCI)
  - Asynchronous Mode/Clock Synchronous Mode
  - 3 Transfer Formats (Asynchronous Mode)
  - 6 Clock Sources
- Memory Ready for Low Speed Memory Access
- Halt
- Error-Detection (Address Error, Op-code Error)
- Interrupts — 3 External, 7 Internal
- Operation Mode
  - MCU Mode
    - Mode 1 — Expanded (Internal ROM Inhibited)
    - Mode 2 — Expanded (Internal ROM Valid)
    - Mode 3 — Single-chip Mode
  - PROM Mode
- Up to 65k Bytes of Address Space
- Low Power Dissipation Mode
  - Sleep
  - Standby
- Minimum Instruction Execution Time — 0.5μs (f=2.0MHz)
- Wide Operation Range
  - V_{CC}=5V±10%
    - f=0.1 to 1.0MHz; HD63701X0
    - f=0.1 to 1.5MHz; HD637A01X0
    - f=0.1 to 2.0MHz; HD637B01X0

■ PIN ARRANGEMENT



(Top View)

■ BLOCK DIAGRAM



■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC}	-0.3 ~ +7.0	V
Program Voltage	V_{PP}	-0.3 ~ 22	V
Input Voltage	V_{in}	-0.3 ~ V_{CC}+0.3	V
Operating Temperature	T_{opr}	0 ~ +70	°C
Storage Temperature	T_{stg}	-55 ~ +125	°C

(Note) This product has protection circuits in input terminal from high static electricity voltage and high electric field. But be careful not to apply overvoltage more than maximum ratings to these high input impedance protection circuits. To assure the normal operation, we recommend V_{in}, V_{out}: $V_{SS} \leqq (V_{in} \text{ or } V_{out}) \leqq V_{CC}$.

■ MCU ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC} =5V±10%, V_{SS} = V_{PP} =0V, Ta= 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage	$\overline{RES}$, $\overline{STBY}$, MP_0, MP_1	V_{IH}		V_{CC}-0.5	–	V_{CC}+0.3	V
	EXTAL			V_{CC}x0.7	–		
	P_{22} (SCLK)***			2.4	–		
	Other Inputs			2.2	–		
Input "Low" Voltage	All Inputs	V_{IL}		-0.3	–	0.8	V
Input Leakage Current	$\overline{NMI}$, $\overline{RES}$, $\overline{STBY}$, MP_0, MP_1, Port 5	$\|I_{in}\|$	V_{in}=0.5 ~ V_{CC}-0.5V	–	–	1.0	μA
Three State (off-state) Leakage Current	Ports 1, 2, 3, 4, 6, 7	$\|I_{TSI}\|$	V_{in}=0.5 ~ V_{CC}-0.5V	–	–	1.0	μA
Output "High" Voltage	All Outputs	V_{OH}	I_{OH}=-200μA	2.4	–	–	V
			I_{OH}=-10μA	V_{CC}-0.7	–	–	V
Output "Low" Voltage	Ports 2, 6	V_{OL}	I_{OL}=1.6mA	–	–	0.5	V
	Other Outputs			–	–	0.4	V
Darlington Drive Current	Ports 2, 6	$-I_{OH}$	V_{out}=1.5V	1.0	–	10.0	mA
Input Capacitance	All Inputs (Except V_{PP} /$\overline{OE}$)	C_{in}	V_{in}=0V, f=1MHz, Ta=25°C	–	–	12.5	pF
	V_{PP} /$\overline{OE}$			–	–	25	pF
Standby Current	Non Operation	I_{STB}		–	3.0	15.0	μA
Current Dissipation*		I_{SLP}	Sleeping (f=1MHz**)	–	1.5	3.0	mA
			Sleeping (f=1.5MHz**)	–	2.3	4.5	mA
			Sleeping (f=2MHz**)	–	3.0	6.0	mA
		I_{CC}	Operating (f=1MHz**)	–	7.0	10.0	mA
			Operating (f=1.5MHz**)	–	10.5	15.0	mA
			Operating (f=2MHz**)	–	14.0	20.0	mA
RAM Standby Voltage		V_{RAM}		2.0	–	–	V

*V_{IH} min = V_{CC}-1.0V, V_{IL} max = 0.8V (All output terminals are at no load.)

**Current Dissipation of the operating or sleeping condition is proportional to the operating frequency. So the typ. or max. values about Current Dissipations at x MHz operation are decided according to the following formula;

typ. value (f = x MHz) = typ. value (f = 1MHz) x x
max. value (f = x MHz) = max. value (f = 1MHz) x x

***Synchronous clock input use only.

• AC CHARACTERISTICS (V_{CC}=5V±10%, V_{SS}=V_{PP}=0V, Ta=0 ~ +70°C, unless otherwise noted.)

BUS TIMING

Item		Symbol	Test Condition	HD63701X0			HD637A01X0			HD637B01X0			Unit
				min	typ	max	min	typ	max	min	typ	max	
Cycle Time		t_{cyc}	Fig. 1	1	–	10	0.666	–	10	0.5	–	10	μs
Enable Rise Time		t_{Er}		–	–	25	–	–	25	–	–	25	ns
Enable Fall Time		t_{Ef}		–	–	25	–	–	25	–	–	25	ns
Enable Pulse Width "High" Level*		PW_{EH}		450	–	–	300	–	–	220	–	–	ns
Enable Pulse Width "Low" Level*		PW_{EL}		450	–	–	300	–	–	220	–	–	ns
Address, R/$\overline{W}$ Delay Time*		t_{AD}		–	–	250	–	–	190	–	–	160	ns
Data Delay Time	Write	t_{DDW}		–	–	200	–	–	160	–	–	120	ns
Data Set-up Time	Read	t_{DSR}		80	–	–	70	–	–	70	–	–	ns
Address, R/$\overline{W}$ Hold Time*		t_{AH}		70	–	–	45	–	–	30	–	–	ns
Data Hold Time	Write*	t_{HW}		70	–	–	50	–	–	35	–	–	ns
	Read	t_{HR}		0	–	–	0	–	–	0	–	–	ns
$\overline{RD}$, $\overline{WR}$ Pulse Width*		PW_{RW}		450	–	–	300	–	–	220	–	–	ns
$\overline{RD}$, $\overline{WR}$ Delay Time		t_{RWD}		–	–	40	–	–	40	–	–	40	ns
$\overline{RD}$, $\overline{WR}$ Hold Time		t_{HRW}		–	–	30	–	–	30	–	–	25	ns
$\overline{LIR}$ Delay Time		t_{DLR}		–	–	200	–	–	160	–	–	120	ns
$\overline{LIR}$ Hold Time		t_{HLR}		10	–	–	10	–	–	10	–	–	ns
MR Set-up Time*		t_{SMR}	Fig. 2	400	–	–	280	–	–	230	–	–	ns
MR Hold Time*		t_{HMR}		–	–	90	–	–	40	–	–	0	ns
E Clock Pulse Width at MR		PW_{EMR}		–	–	9	–	–	9	–	–	9	μs
Processor Control Set-up Time		t_{PCS}	Fig. 3, 10, 11	200	–	–	200	–	–	200	–	–	ns
Processor Control Rise Time		t_{PCr}	Fig. 2, 3	–	–	100	–	–	100	–	–	100	ns
Processor Control Fall Time		t_{PCf}		–	–	100	–	–	100	–	–	100	ns
BA Delay Time		t_{BA}	Fig. 3	–	–	250	–	–	190	–	–	160	ns
Oscillator Stabilization Time		t_{RC}	Fig. 11	20	–	–	20	–	–	20	–	–	ms
Reset Pulse Width		PW_{RST}		3	–	–	3	–	–	3	–	–	t_{cyc}

* These timings change in approximate proportion to t_{cyc}. The figures in this characteristics represent those when t_{cyc} is minimum (= in the highest speed operation).

PERIPHERAL PORT TIMING

Item		Symbol	Test Condition	HD63701X0			HD637A01X0			HD637B01X0			Unit
				min	typ	max	min	typ	max	min	typ	max	
Peripheral Data Set-up Time	Ports 2, 3, 5, 6	t_{PDSU}	Fig. 5	200	–	–	200	–	–	200	–	–	ns
Peripheral Data Hold Time	Ports 2, 3, 5, 6	t_{PDH}	Fig. 5	200	–	–	200	–	–	200	–	–	ns
Delay Time (Enable Negative Transition to Peripheral Data Valid)	Ports 1, 2, 3, 4, 6, 7	t_{PWD}	Fig. 6	–	–	300	–	–	300	–	–	300	ns

TIMER, SCI TIMING

Item		Symbol	Test Condition	HD63701X0			HD637A01X0			HD637B01X0			Unit
				min	typ	max	min	typ	max	min	typ	max	
Timer 1 Input Pulse Width		t_{PWT}	Fig. 8	2.0	–	–	2.0	–	–	2.0	–	–	t_{cyc}
Delay Time (Enable Positive Transition to Timer Output)		t_{TOD}	Fig. 7	–	–	400	–	–	400	–	–	400	ns
SCI Input Clock Cycle	Async. Mode	t_{Scyc}	Fig. 8	1.0	–	–	1.0	–	–	1.0	–	–	t_{cyc}
	Clock Sync.		Fig. 4, 8	2.0	–	–	2.0	–	–	2.0	–	–	t_{cyc}
SCI Transmit Data Delay Time (Clock Sync. Mode)		t_{TXD}	Fig. 4	–	–	200	–	–	200	–	–	200	ns
SCI Receive Data Set-up Time (Clock Sync. Mode)		t_{SRX}		290	–	–	290	–	–	290	–	–	ns
SCI Receive Data Hold Time (Clock Sync. Mode)		t_{HRX}		100	–	–	100	–	–	100	–	–	ns
SCI Input Clock Pulse Width		t_{PWSCK}	Fig. 8	0.4	–	0.6	0.4	–	0.6	0.4	–	0.6	t_{Scyc}
Timer 2 Input Clock Cycle		t_{tcyc}		2.0	–	–	2.0	–	–	2.0	–	–	t_{cyc}
Timer 2 Input Clock Pulse Width		t_{PWTCK}		200	–	–	200	–	–	200	–	–	ns
Timer 1·2, SCI Input Clock Rise Time		t_{CKr}		–	–	100	–	–	100	–	–	100	ns
Timer 1·2, SCI Input Clock Fall Time		t_{CKf}		–	–	100	–	–	100	–	–	100	ns

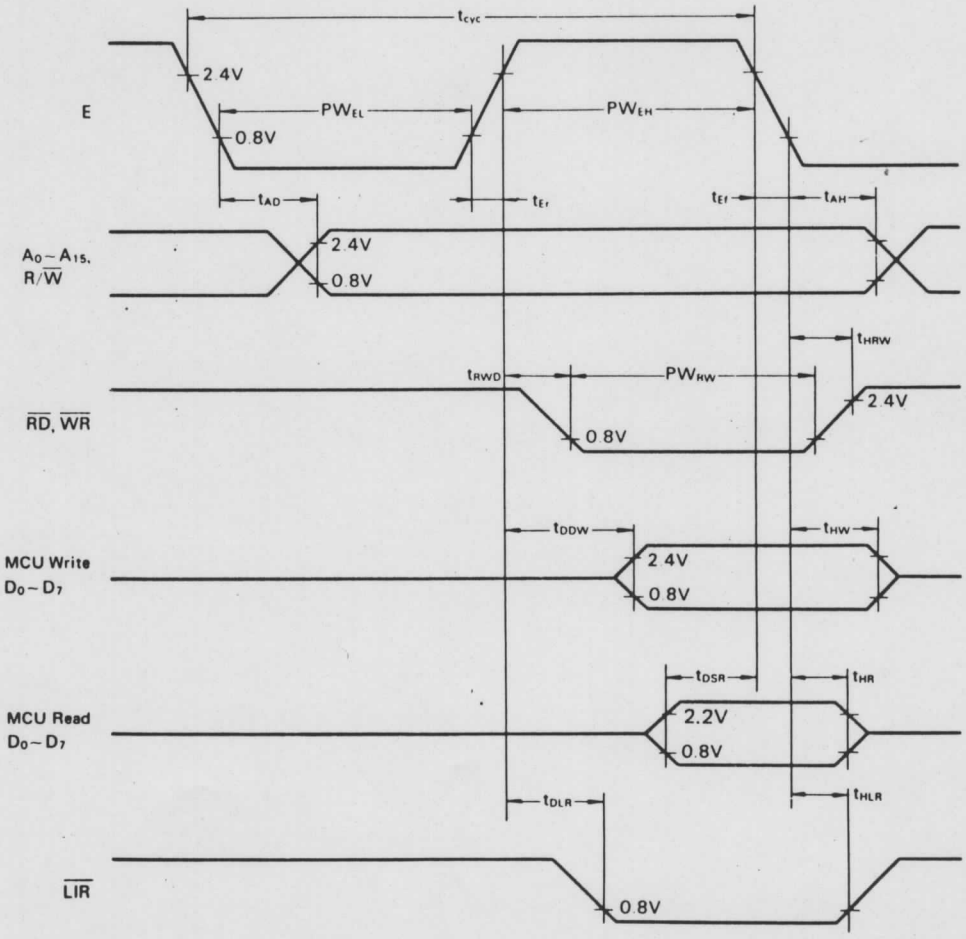


Figure 1 Mode 1, Mode 2 Bus Timing

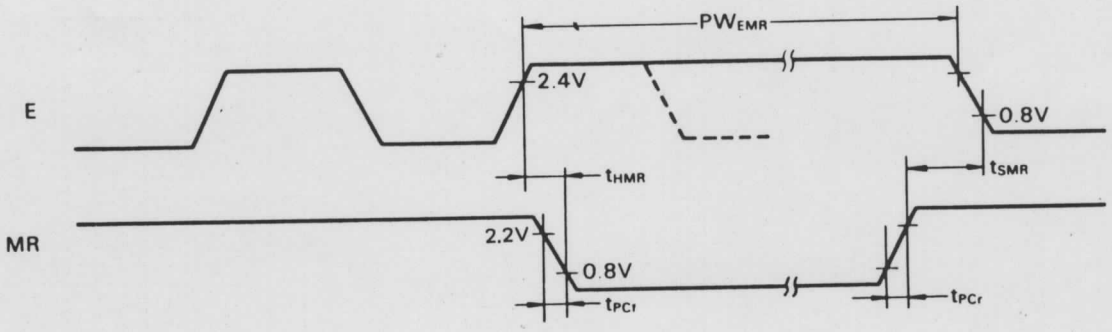


Figure 2 Memory Ready and E Clock Timing

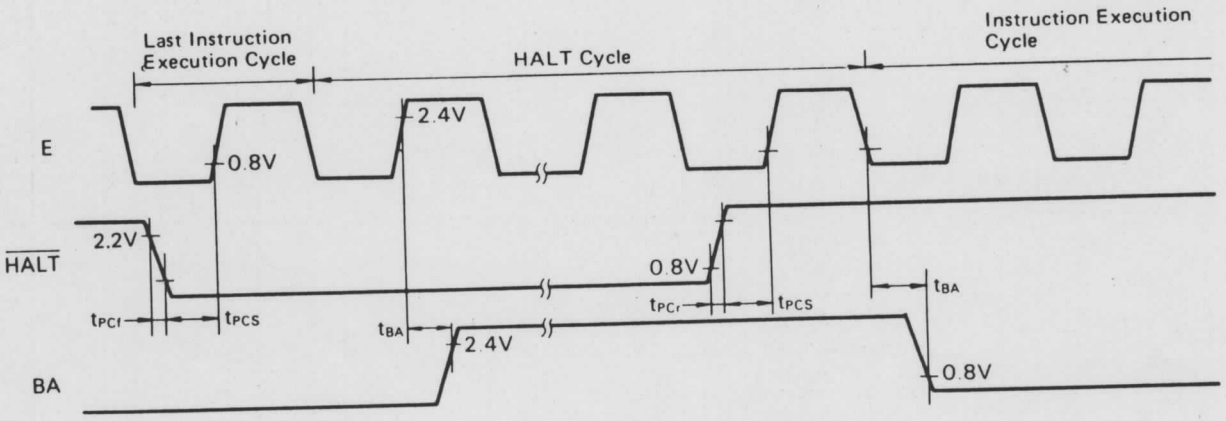


Figure 3 $\overline{\text{HALT}}$ and BA Timing

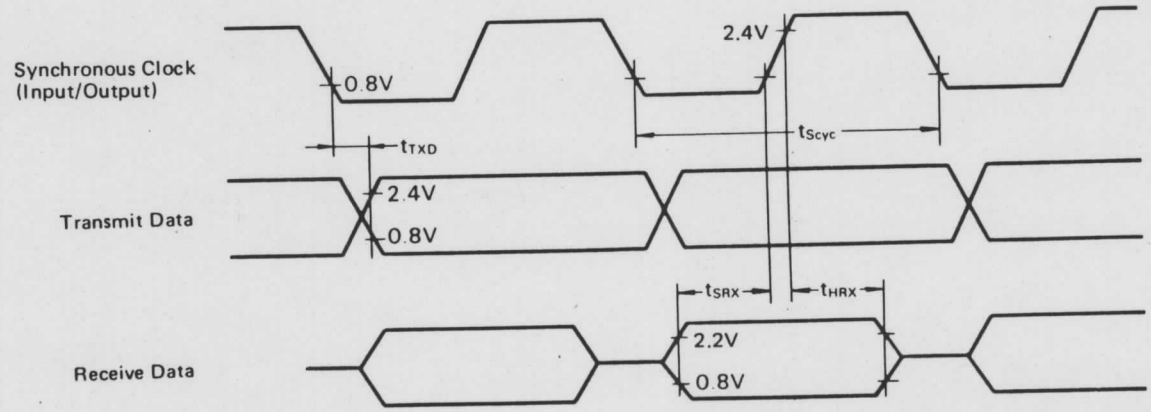


Figure 4 SCI Clocked Synchronous Timing

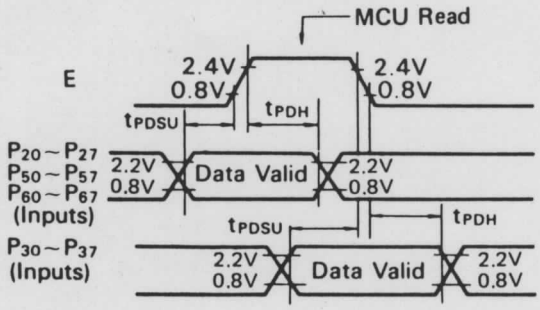


Figure 5 Port Data Set-up and Hold Times (MCU Read)

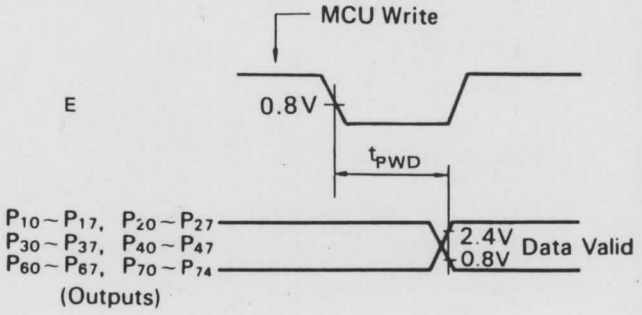
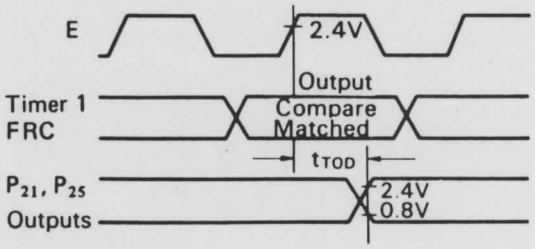
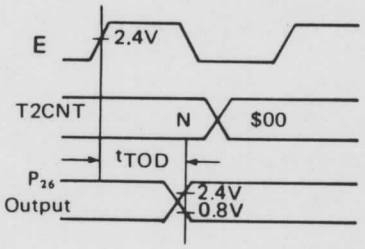


Figure 6 Port Data Delay Times (MCU Write)

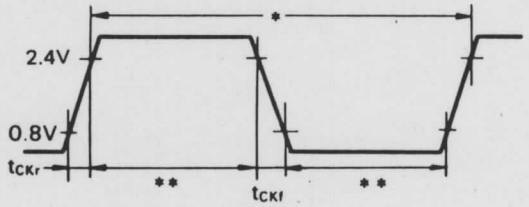


(a) Timer 1 Output Timing



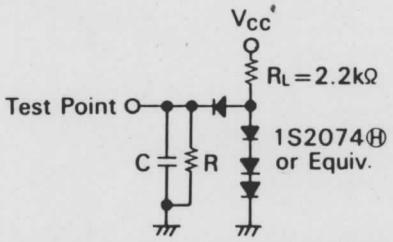
(b) Timer 2 Output Timing

Figure 7 Timer Output Timing



* Timer 2; t_{TCYC} ** Timer 1; t_{PWT}
SCI ; t_{SCYC} Timer 2; t_{PWTCK}
SCI ; t_{PWSCK}

Figure 8 Timer 1·2, SCI Input Clock Timing



C = 90pF for Port 1, Port 3, Port 4, E
= 30pF for Port 2, Port 6, Port 7
R = 12kΩ for Port 1 ~ Port 4, Port 6, Port 7, E

Figure 9 Bus Timing Test Loads (TTL Load)

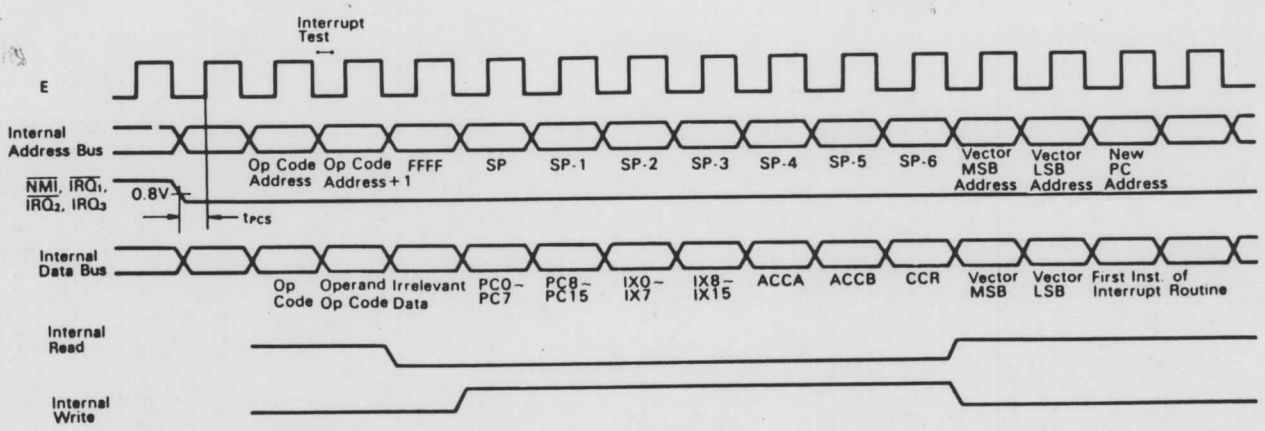


Figure 10 Interrupt Sequence

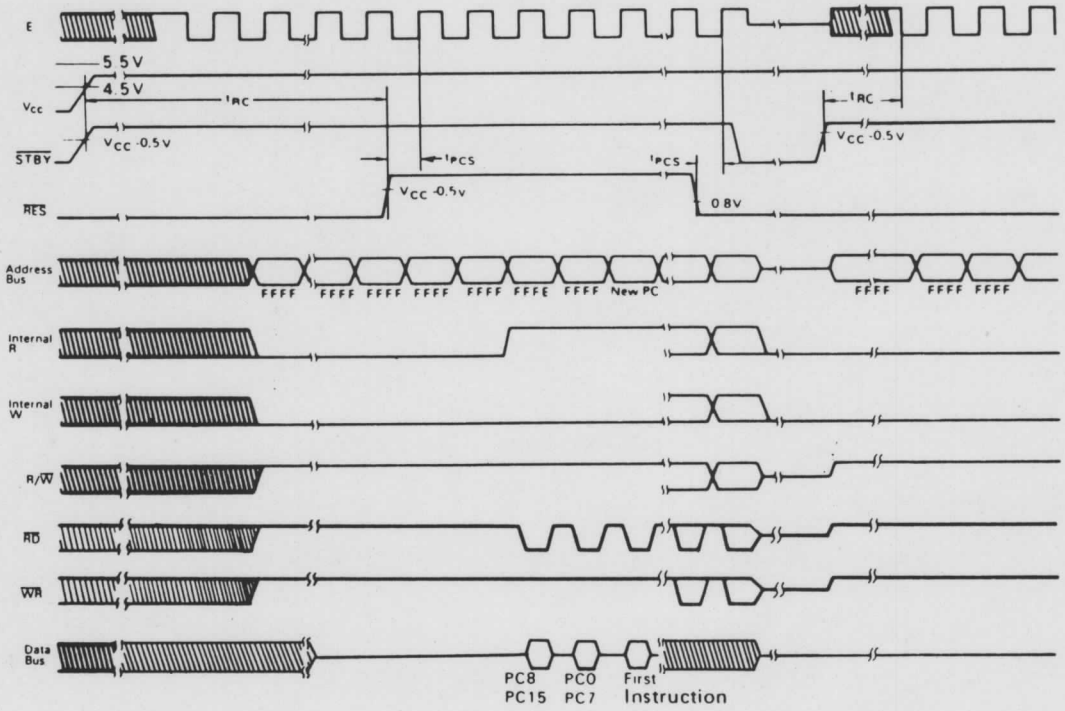


Figure 11 Reset Timing

■ PROM PROGRAMMING ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC}=5V±10%,V_{PP}=21V±0.5V, V_{SS}=0V, Ta=25°C±5°C, unless otherwise noted.)

Item	Symbol	Test Condition	min	typ	max	Unit
Program Voltage	V_{PP}		20.5	21	21.5	V
Program Current	I_{PP}	V_{PP}=21V, $\overline{CE}$=V_{IL}	–	–	30	mA
Input Leakage Current	I_{LI}	V_{in}=5.25V/0.4V	–	–	10	μA
Input "Low" Voltage	V_{IL}		-0.1	–	0.8	V
Input "High" Voltage	V_{IH}		2.2	–	V_{CC}+1.0	V
Output "Low" Voltage	V_{OL}	I_{OL}=1.6mA	–	–	0.45	V
Output "High" Voltage	V_{OH}	I_{OH}=-200 μA	2.4	–	–	V

• AC CHARACTERISTICS (V_{CC}=5V±10%,V_{PP}=21V±0.5V, V_{SS}=0V, Ta=25°C±5°C, unless otherwise noted.)

Item	Symbol	Test Condition	min	typ	max	Unit
Address Set-up Time	t_{AS}		2	–	–	μs
Address Hold Time	t_{AH}		0	–	–	μs
$\overline{OE}$ Set-up Time	t_{OES}		2	–	–	μs
$\overline{OE}$ Hold Time	t_{OEH}		2	–	–	μs
Data Set-up Time	t_{DS}		2	–	–	μs
Data Hold Time	t_{DH}		2	–	–	μs
Output Disable Delay Time	t_{DF}		0	–	130	ns
Data Valid from $\overline{CE}$	t_{DV}	$\overline{CE}$=V_{IL}, $\overline{OE}$=V_{IL}	–	–	1	μs
$\overline{CE}$ Pulse Width	t_{PW}		45	50	55	ms
$\overline{OE}$ Pulse Rise Time	t_{PRT}		50	–	–	ns
V_{pp} Recovery Time	t_{VR}		2	–	–	μs

(Note) t_{DF} is defined when output becomes open because output level can not be refered.

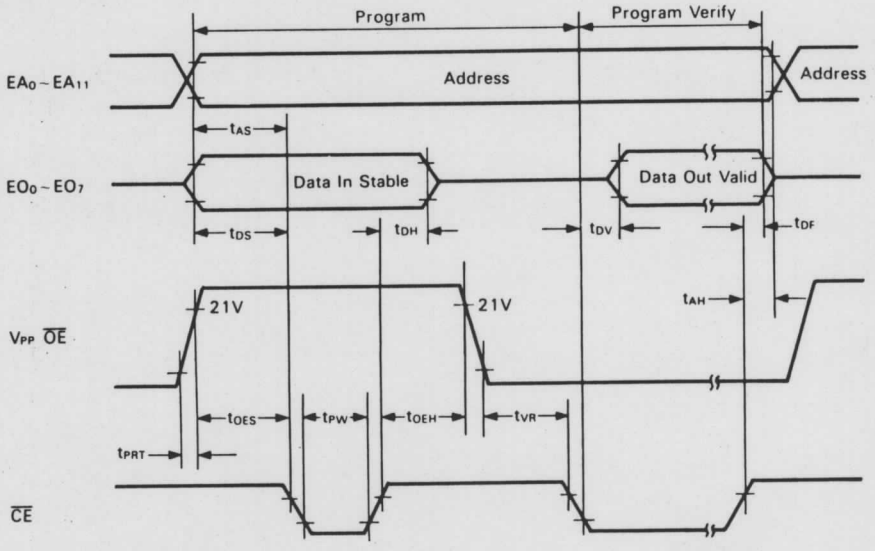


Figure 12 PROM Programming Timing

Test Condition;
Input pulse level ..... 0.8V to 2.2V
Input rise/fall time ..... ≦ 20 ns
I/O timing reference level..... input 1V, 2V
output 0.8V, 2V

■ FUNCTIONAL PIN DESCRIPTION

• V_{CC}, V_{SS}

V_{CC} and V_{SS} provide power to the MCU with 5V±10% supply. V_{SS} pin should be tied to ground.

• XTAL, EXTAL

These two pins interface a crystal (an AT-cut type). Divide-by-four circuit is on chip. When 4MHz crystal is used, the system clock is 1MHz for example.

EXTAL pin may be driven with an external clock of 45 to 55% duty, and one fourth frequency of the external clock is produced in the LSI. The external clock frequency should be less

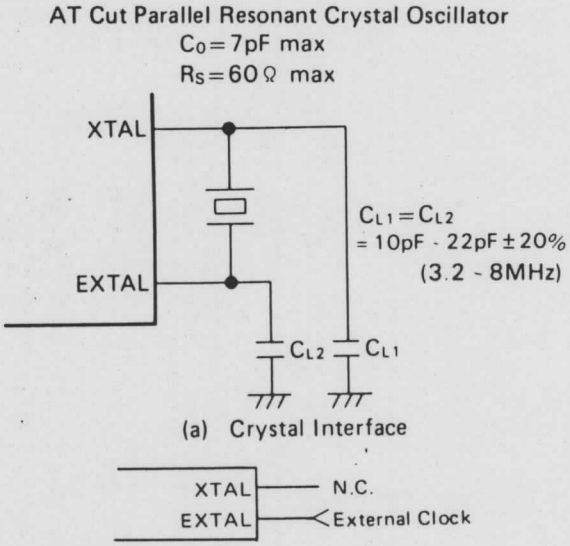


Figure 13 Connection Circuit

than four times of the maximum frequency. When using the external clock, XTAL pin should be open. Fig. 13 shows an example of connection circuit. The crystal and C_{L1}, C_{L2} should be mounted as close as possible to XTAL and EXTAL pins. Any line must not cross the line between the crystal and XTAL, EXTAL.

• $\overline{STBY}$

This pin is used for standby mode or PROM mode.

In standby mode, the oscillation may be stopped. To retain the contents of RAM at standby, "0" should be written into RAM enable bit (RAMW). RAME is the bit 6 of the RAM/Port 5 control register at $0014. RAM is disabled by this operation and its contents is sustained. Refer to "LOW POWER DISSIPATION MODE" for standby mode.

When this pin and Mode Program pins, MP_0 and MP_1, are "Low" level, the MCU is in PROM mode. Refer to "THE PROM PROGRAMMING" for details.

• Reset ($\overline{RES}$)

This pin is used to reset the MCU's internal state and provide a startup procedure. During power up, $\overline{RES}$ pin must be held below "Low" level for more than 20 ms.

The CPU registers (accumulator, index register, stack pointer, condition code register except for interrupt mask bit), RAM and data registers of ports are not initialized during reset, so their contents are unknown in a startup procedure.

To reset the MCU during operation, $\overline{RES}$ should be held "Low" for at least 3 system-clock cycles. At the 3rd cycle, all the address buses become "High". When $\overline{RES}$ remains "Low", the address buses keep "High". If $\overline{RES}$ turns "High", the MCU restart sequence is:

(1) Latch the value of the mode program pins: MP_0 and MP_1.
(2) Initialize each internal register (refer to Table 5).
(3) Set the interrupt mask bit. For the CPU to recognize the maskable interrupts $\overline{IRQ_1}$, $\overline{IRQ_2}$ and IRQ_3, this bit should be cleared in advance.
(4) Put the contents (=start address) of the last two addresses ($FFFE, $FFFF) into the program counter and start the program from this address. (Refer to Table 1).

* The MCU is unable to accept a reset input until the clock becomes normal oscillation after power on (max. 20ms). During this transient time, the MCU and I/O pins are undefined. Please be aware of this for system designing.

• Enable (E)

This pin provides a TTL-compatible clock used for bus synchronization. Its frequency is one fourth that of the internal oscillator or external clock. This pin can drive one TTL load and 90pF capacitance.

• Non-Maskable Interrupt ($\overline{NMI}$)

When the negative edge of the input signal is detected at this pin, the CPU will begin a non-maskable interrupt sequence. But the current instruction will be completed before it responds to the request. The interrupt mask bit of the condition code register doesn't affect non-maskable interrupt at all.

When the interrupt occurs, the contents of the program counter, the index register, the accumulators and the condition code register will be pushed onto the stack. Upon completion of this sequence, a vector is fetched from $FFFC and $FFFD, transferred their contents to the program counter and the non-maskable interrupt service routine starts. After reset, the stack pointer should be initialized on an appropriate memory area before $\overline{NMI}$ input.

• Interrupt Request ($\overline{IRQ_1}$, $\overline{IRQ_2}$)

These are level-sensitive pins which request an internal interrupt sequence. At interrupt request, the CPU will complete the current instruction before it responds to the request. If the interrupt mask in the condition code register is clear, the CPU will begin an interrupt sequence; if set, the interrupt request will be ignored. When the sequence starts, the contents of the program counter, the index register, the accumulators and the condition code register will be pushed onto the stack, then the interrupt mask bit will be set and inhibits all maskable interrupt. Finally, a vector is fetched from an address depicted in Table 1 and transferred to the program counter, and instruction execution is resumed.

The external interrupt pins, $\overline{IRQ_1}$ and $\overline{IRQ_2}$ are also used for port pins P_{50} and P_{51}, so it is controlled by Bit 0 and 1 of the RAM/Port 5 control register at $0014. Refer to "RAM/PORT 5 CONTROL REGISTER" for details.

One of the internal interrupts, ICI, OCI, TOI, CMI or SIO can generate an internal interrupt (IRQ_3). IRQ_3 function is just the same as $\overline{IRQ_1}$ or $\overline{IRQ_2}$ except the vector address. Fig. 14 shows the block diagram of the interrupt circuit.

• Mode Program (MP_0, MP_1)

These two pins decide the operation mode. Refer to "MODE SELECTION" for more details.

Table 1 Interrupt Vector Memory Map

Priority	Vector		Interrupt
	MSB	LSB	
Highest	FFFE	FFFF	$\overline{RES}$
	FFEE	FFEF	TRAP
	FFFC	FFFD	$\overline{NMI}$
	FFFA	FFFB	SWI (Software Interrupt)
	FFF8	FFF9	$\overline{IRQ_1}$
	FFF6	FFF7	ICI (Timer 1 Input Capture)
	FFF4	FFF5	OCI (Timer 1 Output Compare 1, 2)
	FFF2	FFF3	TOI (Timer 1 Overflow)
	FFEC	FFED	CMI (Timer 2 Counter Match)
	FFEA	FFEB	$\overline{IRQ_2}$
Lowest	FFF0	FFF1	SIO (RDRF+ORFE+TDRE)

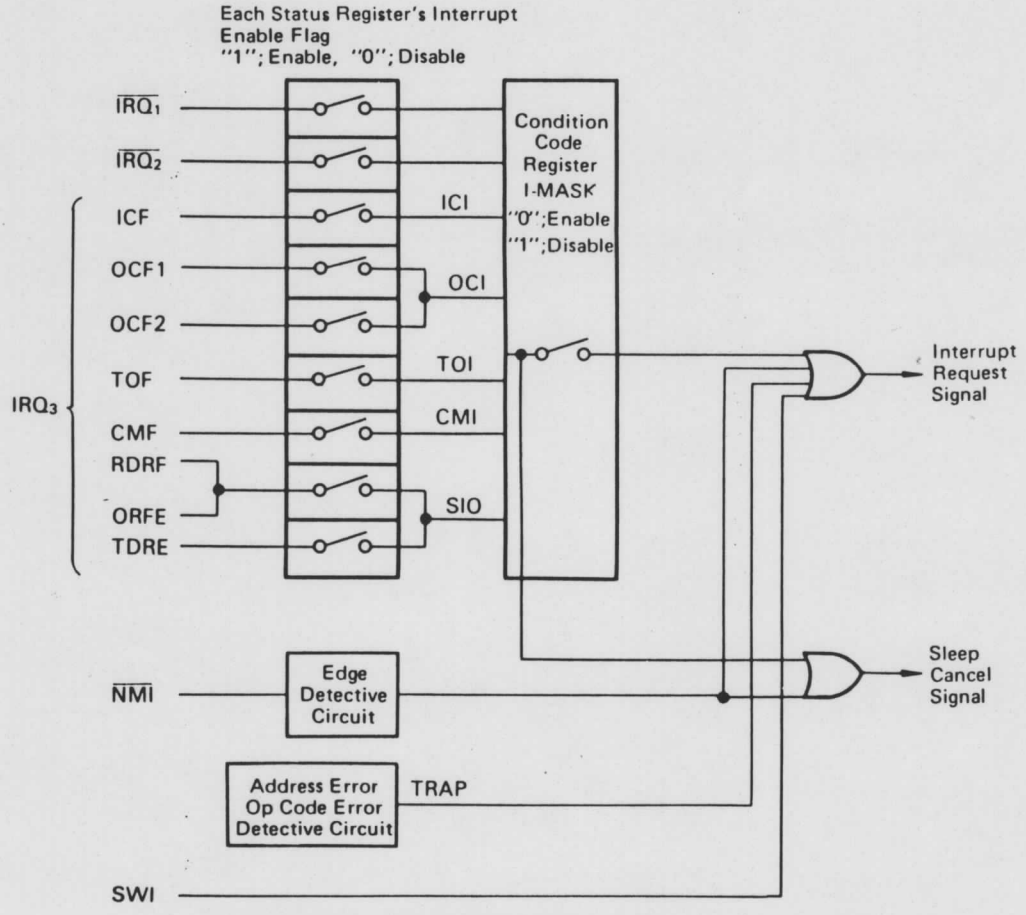


Figure 14 Interrupt Circuit Block Diagram

The following signal descriptions are applied only for expanded mode.

- **Read/Write (R/$\overline{W}$; P_{72})**

This signal, usually in read state ("High"), shows whether the MCU is in read ("High") or write ("Low") state. This can drive one TTL load and 30pF capacitance.

- **$\overline{RD}$, $\overline{WR}$ (P_{70}, P_{71})**

These outputs will turn "Low" when the CPU read/write operation is completed. This enables the CPU easy to access the peripheral LSI with $\overline{RD}$ and $\overline{WR}$ input pins. These pins can drive one TTL load and 30pF capacitance.

- **Load Instruction Register ($\overline{LIR}$; P_{73})**

This is output for the instruction opecode on data bus (active low). This pin can drive one TTL load and 30pF capacitance.

- **Memory Ready (MR; P_{52})**

This input is used to stretch the system clock's "High" period in order to access low-speed memories. During this signal being in "High", the system clock operates in normal sequence. But in "Low", the "High" period of the system clock will be stretched in integral multiples of the cycle time. This allows the CPU to interface with low-speed memories (See Fig. 2). Up to 9μs can be stretched.

During internal address access or nonvalid memory access, MR is prohibited internally to prevent decrease of operation speed. Even in the halt state, MR can also stretch "High" period of system clock to allow peripheral devices to access low-speed memories. As this pin is used also for P_{52}, an enable bit is provided at bit 2 of the RAM/Port 5 control register at $0014. Refer to "RAM/PORT 5 CONTROL REGISTER" for more details.

- **Halt ($\overline{HALT}$; P_{53})**

This input is used to stop instruction execution or to release buses free. When this signal turns "Low", the CPU will be in the halt state after completing the current instruction. During the halt state, BA (P_{74}) is in "High", and an address bus, data bus, $\overline{RD}$, $\overline{WR}$ and R/$\overline{W}$ are high impedance. When an interrupt is requested in the halt state, the CPU responds to the interrupt request after the halt is cancelled.

(Note) When the CPU is interrupt wait state in WAI instruction execution, $\overline{HALT}$ should be held "High". If $\overline{HALT}$ turns "Low", the CPU may malfunction after releasing the halt state.
Refer to "APPLICATION NOTES—Precaution for using WAI instruction" for more details.

- **Bus Available (BA; P_{74})**

This output is normally "Low" but "High" when the CPU accepts $\overline{HALT}$ and releases the buses. The HD6800 and HD6802 make BA "High" and release the buses at WAI execution, while the HD63701X0 doesn't make BA "High" under the same condition.

The following pin functions are applied only in PROM mode. Refer to "THE PROM PROGRAMMING" for details of PROM mode.

- **Chip Enable ($\overline{CE}$; P_{57})**

This pin is input for programming and verifying the PROM. When this pin is "Low" level, PROM will be enable.

The PROM can not be programmed or verified in "High" level.

- **Program Voltage/Output Enable (V_{PP}/$\overline{OE}$)**

This pin is used for program voltage and data output control in verification.

Data from Port 3 (EO_0 to EO_7) can be programmed into the PROM when applying 21V±0.5V to V_{PP} and holding $\overline{CE}$ in "Low" level. The PROM address is provided to Port 1 and Port 4 (EA_0 to EA_{11}). In verification, the PROM data is output from Port 3 (EO_0 to EO_7) when this pin is "Low" level. In "High" level, Port 3 will be high-impedance. In MCU mode, this pin should be connected to V_{SS}.

■ PORT

The HD63701X0 has six 8-bit ports and a 5-bit port. Table 2 gives the address of ports and the data direction register and Fig. 15 the block diagrams of each port.

Table 2 Port and Data Direction Register Address

Port	Port Address	Data Direction Register
Port 1	$0002	–
Port 2	$0003	$0001
Port 3	$0006	$0004
Port 4	$0007	–
Port 5	$0015	–
Port 6	$0017	$0016
Port 7	$0018	–

- **Port 1**

In MCU mode, port 1 is used for an 8-bit output port. In mode 3, port 1 is high impedance during reset, and keeps the state even after reset is released. When the CPU writes on the port 1 data register, the written data will appear at Port 1. Once port 1 gets in the output state, it operates as an output till reset. The CPU can read the Port 1 data register for the bit manipulation instruction.

In mode 1 and 2, port 1 is used for lower address buses. This port can drive one TTL load and 90pF capacitance.

In PROM mode, port 1 is lower address bus (EA_0 to EA_7) for the PROM.

- **Port 2**

An 8-bit input/output port. Its I/O state depends on the data direction register (DDR) of port 2 which provides two bits; bit 0 decides the I/O direction of P_{20} and bit 1 the I/O direction of P_{21} to P_{27} ("0" for input, "1" for output).

Port 2 is also used for the timers and the SCI. When used for the timers and the SCI, P_{21} to P_{27} are decided I/O regardless of the DDR (except for P_{20}).

Port 2 Data Direction Register

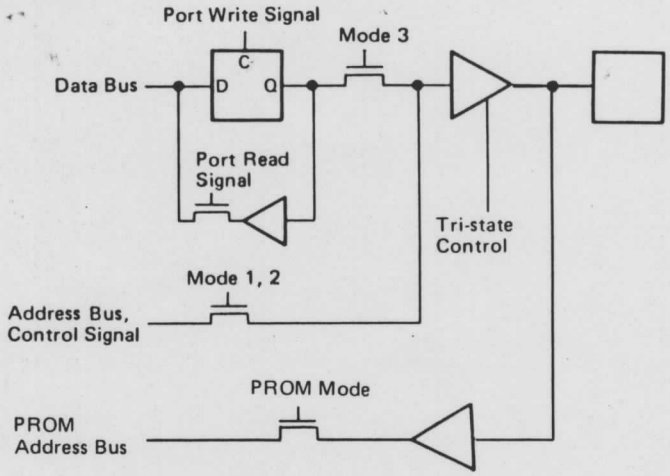
7	6	5	4	3	2	1	0	
–	–	–	–	–	–	DDR 1–7	DDR 0	$0001

The DDR of port 2 is cleared at reset and port 2 is configured as an input. This port can drive one TTL and 30pF. In addition, it is capable of sinking 1mA current at Vout=1.5V to drive directly the base of Darlington transistors.

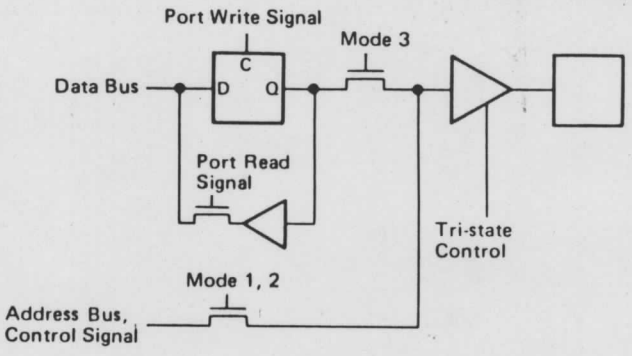
- **Port 3**

An 8-bit I/O port. I/O state depends on the DDR of Port 3 which has only one bit ("0" for input and "1" for output). It is cleared at reset. In mode 1 and 2, port 3 is used for data bus. This port can drive one TTL load and 90pF capacitance.

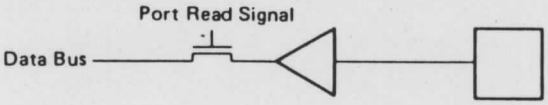
Port 3 is used for data bus (EO_0 to EO_7) of PROM in PROM mode. In this case, I/O state of Port 3 is selected by $\overline{OE}$ but not the DDR.



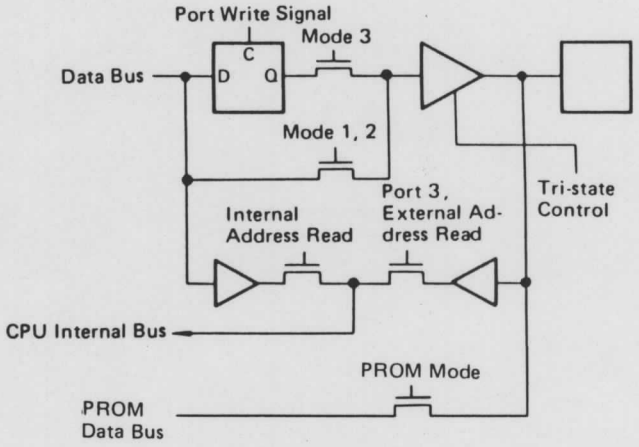
Port 1, Port 4 (Bit 0 to 3)



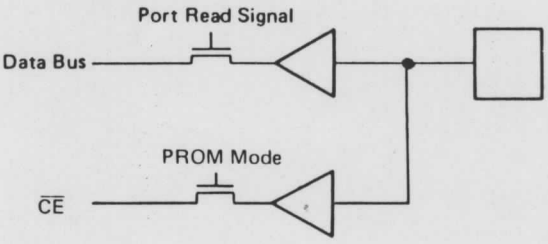
Port 4 (Bit 4 to 7), Port 7



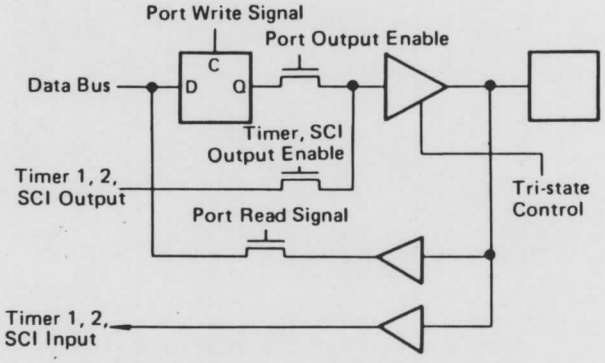
Port 5



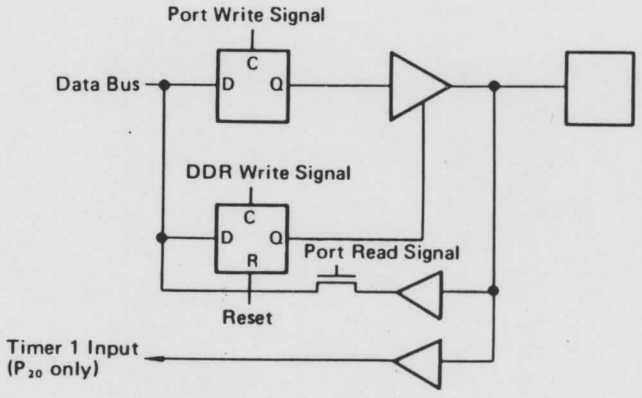
Port 3



Port 5 (Bit 7)



Port 2



Port 6, Port 2 (Bit 0)

Figure 15 Port Block Diagram

Port 3 Data Direction Register

7	6	5	4	3	2	1	0	
–	–	–	–	–	–	–	Port 3 DDR	$0004

- **Port 4**

In MCU mode, Port 4 is used for an 8-bit output port like Port 1. In mode 1 and 2, it is used for upper address bus.

In PROM mode, P_{40} to P_{43} are used for upper address bus (EA_8 to EA_{11}) of PROM.

- **Port 5**

An 8-bit input port. The lower 4 bits are used for interrupt, MR, $\overline{HALT}$, and P_{57} is $\overline{CE}$ for the PROM control.

- **Port 6**

An 8-bit I/O port. This port is programmable as either input or output under software control of the corresponding the DDR ("0" for input, "1" for output). This port can drive one TTL load and 30pF. The DDR of Port 6 is cleared at reset. In addition, it is capable to sinking 1mA current at Vout=1.5V to drive directly the base of Darlington transistors.

- **Port 7**

A 5-bit output port. In mode 3, Port 7 is high impedance during reset and keeps the state even after reset is released. When the CPU writes on the Port 7 data register, the written data will appear at Port 7. Once Port 7 gets in the output state, it operates as an output till reset. The CPU can read the data register for the bit manipulation instruction. In this case b_7 to b_5 are "1".

In mode 1 and 2, Port 7 is used for control signals ($\overline{RD}$, $\overline{WR}$, R/$\overline{W}$, $\overline{LIR}$ and BA). This port can drive one TTL load and 30pF.

■ RAM/PORT 5 CONTROL REGISTER

The control register located at $0014 controls on-chip RAM and Port 5.

RAM/Port 5 Control Register

7	6	5	4	3	2	1	0	
STBY PWR	RAME	–	–	HLTE	MRE	IRQ_2 E	IRQ_1 E	$0014

Bit 0, Bit 1 $\overline{IRQ_1}$, $\overline{IRQ_2}$ Enable Bit (IRQ_1E, IRQ_2E)

When using P_{50} and P_{51} for interrupt pins, write "1" in these bits. When "0", the CPU doesn't accept an external interrupt or a sleep cancellation by the external interrupt. These bits are cleared at reset.

Bit 2 Memory Ready Enable Bit (MRE)

When using P_{52} for an input for Memory Ready signal, write "1" in this bit. When "0", the memory ready function is prohibited and P_{52} is for port. In mode 3, the memory ready function is prohibited regardless of the value of this bit. This bit is set at reset.

Bit 3 Halt Enable Bit (HLTE)

When using P_{53} for an input for Halt signal, write "1" in this bit. When "0", the halt function is prohibited. In mode 3, the halt function is prohibited regardless of the value of this bit. This bit is set at reset.

(Note) When using P_{52} and P_{53} for port in mode 1 and 2, MRE and MLTE must be cleared after reset. If P_{52} or P_{53} turns "Low" before MRE and HLTE are cleared, the memory ready function or the halt function will not be prohibited.

Bit 4, Bit 5 Not Used.

Bit 6 RAM Enable (RAME)

The RAM is controlled by this bit. It is set at reset and the RAM is enabled. This bit is programmable by software. When the RAM is disabled (=logic "0"), the CPU can access an external memory. This bit should be cleared at the beginning of standby mode to protect the RAM data.

Bit 7 Standby Power Bit (STBY PWR)

This bit is cleared whenever V_{CC} decreases below V_{RAM} (min). This is a read/write status bit by software. If this bit is set before standby mode, it indicates that V_{CC} is applied and the RAM is valid.

■ MODE SELECTION

The HD63701X0 provides two fundamental modes, MCU mode and PROM mode. MCU mode is grouped into three; two expanded modes (mode 1, mode 2) and a single chip mode (mode 3).

These operating modes are selectable by mode program pins, MP_0 and MP_1, and standby pin, $\overline{STBY}$ as shown in Table 3.

● Mode 1 (Expanded Mode)

In this mode, Port 3 is data bus, Port 1 is lower address bus and Port 4 is upper address bus to interface with the HMCS6800 buses. Port 7 is used for control signal such as R/$\overline{W}$. In mode 1, the PROM is disable and external address space are expandable up to 65k bytes (refer to Fig. 16).

● Mode 2 (Expanded Mode)

This mode is also expanded mode. But in mode 2, address space is expandable up to 61k bytes and the PROM is enable (refer to Fig. 17).

● Mode 3 (Single-chip Mode)

In this mode, all ports are available (refer to Fig. 18).

● PROM Mode

In this mode, the PROM can be programmed. Refer to "THE PROM PROGRAMMING" for details.

● Mode and Ports

Table 4 shows the MCU signals in each mode.

Table 3 Mode Selection

Mode		MP_1	MP_0	$\overline{STBY}$	PROM	RAM	Interrupt Vector	Operation Mode
MCU Mode	1	"L"	"H"	*	E	I (Note 1)	E	Expanded Mode
	2	"H"	"L"	*	I	I (Note 1)	I	Expanded Mode
	3	"H"	"H"	*	I	I	I	Single-chip Mode
PROM Mode		"L"	"L"	"L"	I	*	*	PROM Programming Mode

"L"=Logic "0", "H"=Logic "1", I; Internal, E; External, *; Don't care
(Note 1) The RAM address area will be external by clearing RAME bit at $0014.

Table 4 MCU Signals in Each Mode

Mode / Port	MCU Mode			PROM Mode
	Mode 1	Mode 2	Mode 3	
Port 1	Address Bus ($A_0 \sim A_7$)	Address Bus ($A_0 \sim A_7$)	Output Port	Address Bus ($EA_0 \sim EA_7$)
Port 2	I/O Port	I/O Port	I/O Port	No use (Note 3)
Port 3	Data Bus ($D_0 \sim D_7$)	Data Bus ($D_0 \sim D_7$)	I/O Port	Data Bus ($EO_0 \sim EO_7$)
Port 4	Address Bus ($A_8 \sim A_{15}$)	Address Bus ($A_8 \sim A_{15}$)	Output Port	Address Bus ($EA_8 \sim EA_{11}$) (Note 1)
Port 5	Input Port	Input Port	Input Port	$\overline{CE}$ (P_{57}) (Note 2)
Port 6	I/O Port	I/O Port	I/O Port	No use (Note 3)
Port 7	$\overline{RD}$, $\overline{WR}$, R/$\overline{W}$, $\overline{LIR}$, BA	$\overline{RD}$, $\overline{WR}$, R/$\overline{W}$, $\overline{LIR}$, BA	Output Port	No use (Note 3)

(Note 1) Use only 4 pins P_{40} to P_{43}. P_{44} to P_{47} are not used.
(Note 2) 7 pins P_{50} to P_{56} are not used.
(Note 3) Unused ports should be connected to V_{SS}.

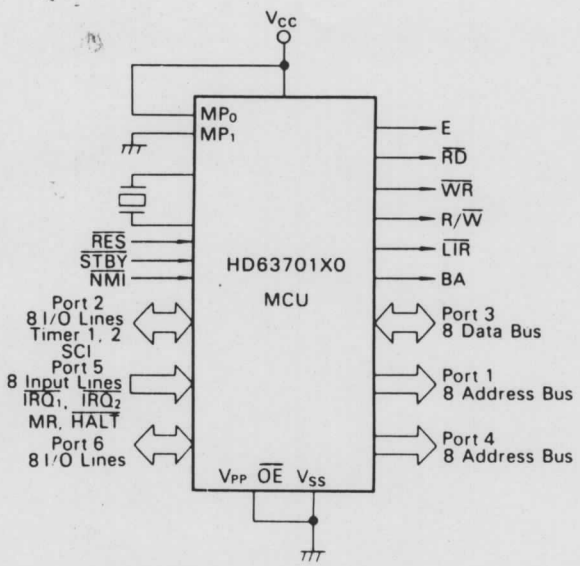


Figure 16 MCU Mode; Mode 1

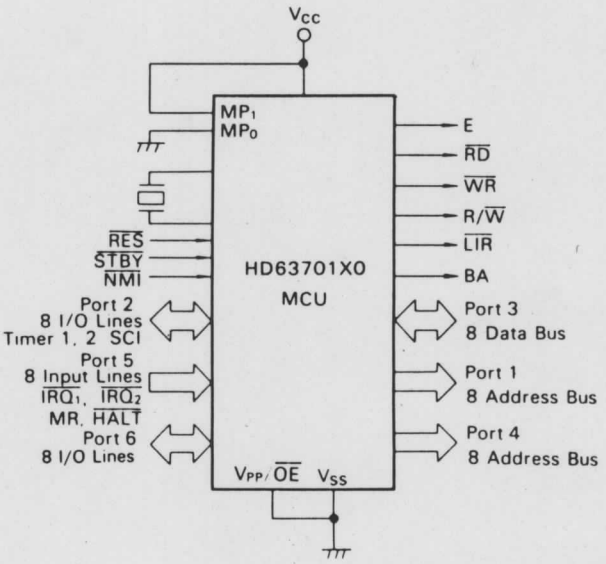


Figure 17 MCU Mode; Mode 2

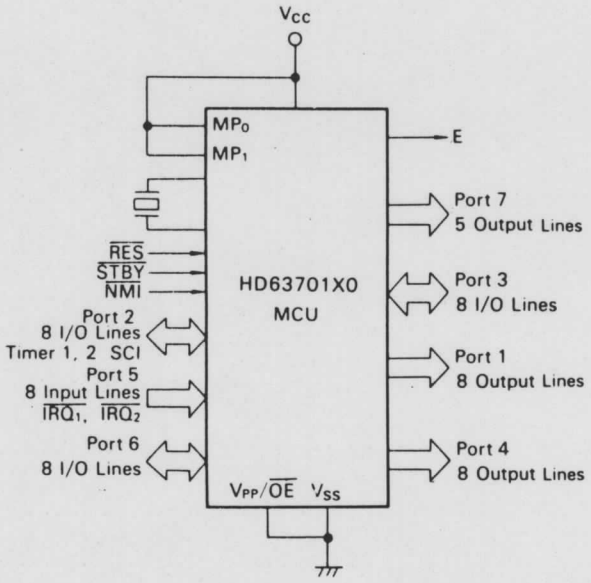


Figure 18 MCU Mode; Mode 3

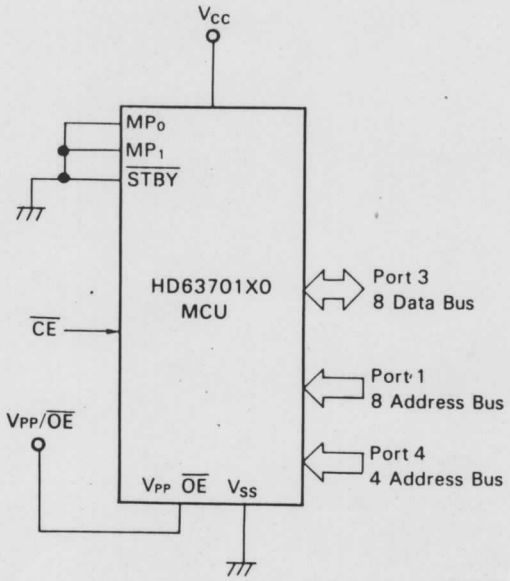


Figure 19 PROM Mode

■ MEMORY MAP

The MCU has ability to access a 65k byte memory space depending on the operating mode. A memory map for each operating mode is shown in Fig. 20. The first 32 locations of each map are reserved for the MCU's internal register area, as shown in Table 5.

Table 5 Internal Register

Address	Registers	R/W***	Initialize at RESET
00	–	–	–
01	Port 2 Data Direction Register	W	$FC
02*	Port 1	R/W	Undefined
03	Port 2	R/W	Undefined
04*	Port 3 Data Direction Register	W	$FE
05	–	–	–
06*	Port 3	R/W	Undefined
07*	Port 4	R/W	Undefined
08	Timer Control/Status Register 1	R/W	$00
09	Free Running Counter ("High")	R/W	$00
0A	Free Running Counter ("Low")	R/W	$00
0B	Output Compare Register 1 ("High")	R/W	$FF
0C	Output Compare Register 1 ("Low")	R/W	$FF
0D	Input Capture Register ("High")	R	$00
0E	Input Capture Register ("Low")	R	$00
0F	Timer Control/Status Register 2	R/W	$10
10	Rate, Mode Control Register	R/W	$00
11	Tx/Rx Control Status Register	R/W	$20
12	Receive Data Register	R	$00
13	Transmit Data Register	W	$00
14	RAM/Port 5 Control Register	R/W	$7C or $FC
15	Port 5	R	–
16	Port 6 Data Direction Register	W	$00
17	Port 6	R/W	Undefined
18*	Port 7	R/W	Undefined
19	Output Compare Register 2 ("High")	R/W	$FF
1A	Output Compare Register 2 ("Low")	R/W	$FF
1B	Timer Control/Status Register 3	R/W	$20
1C	Time Constant Register	W	$FF
1D	Timer 2 Up Counter	R/W	$00
1E	–	–	–
1F**	Test Register	–	–

* External Address in Mode 1, 2.
** Test Register. Do not access to this register.
*** R : Read Only Register
W : Write Only Register
R/W : Read/Write Register

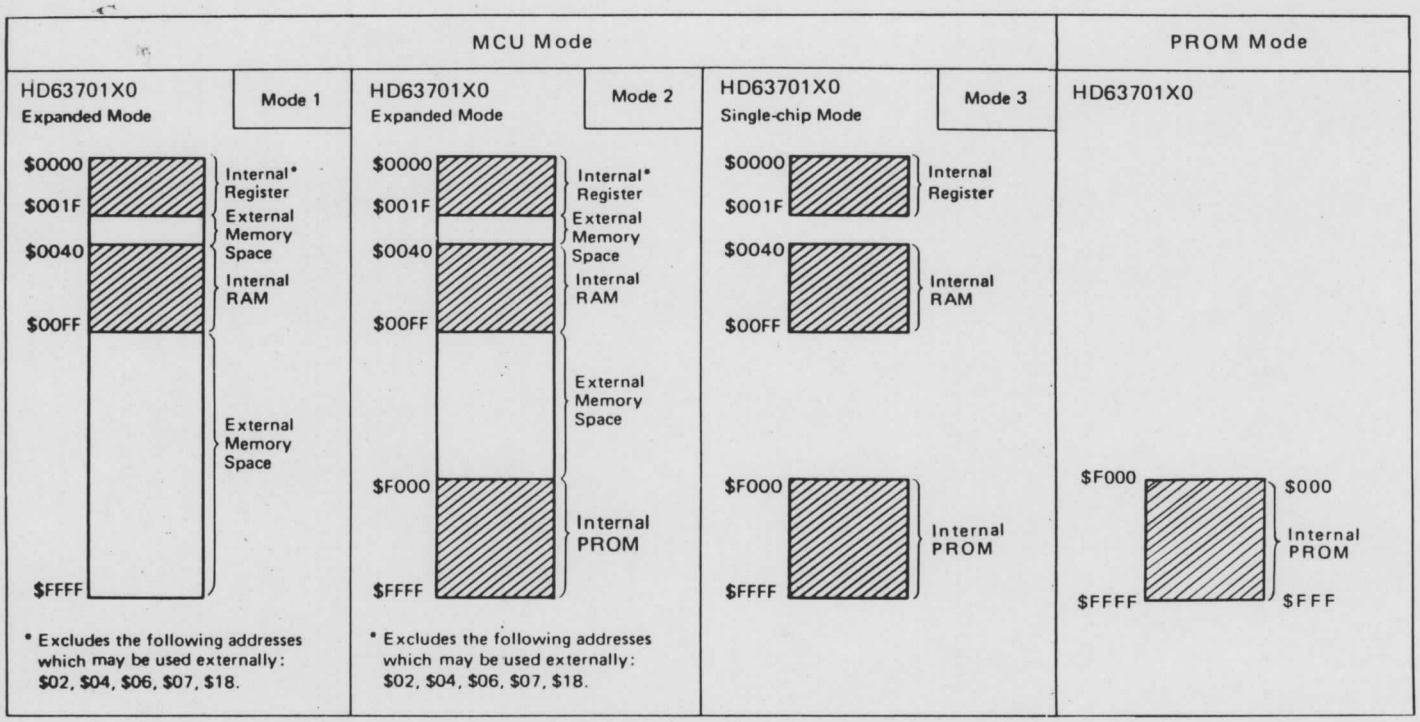


Figure 20 HD63701X0 Memory Map

■ THE PROM PROGRAMMING

The HD63701X0 does not operate as the MCU in PROM mode, which allows to be programmable as equivalent EPROM 2732A type. When three pins, MP_0, MP_1 and $\overline{STBY}$ should be held low, the MCU will be in PROM mode as shown in Table 3. In this mode, P_{30} to P_{37} are used for data bus, P_{10} to P_{17} and P_{40} to P_{43} for address bus, and P_{57} for $\overline{CE}$ input. (refer to Fig. 19.).

● Programming/Verification

When $\overline{CE}$ pin is held low after the program voltage (V_{PP}) is applied to $V_{PP}/\overline{OE}$ pin, the data byte can be applied to Port 3. When $V_{PP}/\overline{OE}$ pin and $\overline{CE}$ pin are held low after programming, the programmed data is output from Port 3 and user can verify the data. I/O timing of these signals are referred to Fig. 12.

When $\overline{CE}$ pin is returned to high, Port 3 will be tri-state and PROM programming/verification will be inhibited.

Table 6 shows the condition of the each pin is PROM mode. Unused pins should be connected to GND in PROM mode.

(Note) It is impossible to erase the programmed PROM of the plastic molded HD63701X0.
Refer to "APPLICATION NOTES—The PROM Programming and Maintenance" for details.

■ TIMER 1

The HD63701X0 has a 16-bit programmable timer which can be used to perform input waveform measurements while generating two independent output waveforms. The pulse width can vary from several microseconds to many seconds.

Timer 1 is configurated as follows (refer to Fig. 22).

Table 6 Pin Condition in PROM mode

Pin / No. / Mode	V_{CC}	V_{SS}	$V_{PP}/\overline{OE}$	$\overline{CE}$	P_{30} to P_{37}	P_{10} to P_{17} P_{40} to P_{43}	MP_0, MP_1, $\overline{STBY}$	Other pins
	33	1	42	24	51 to 58	43 to 50 38 to 41	4, 5, 7	
Programming	+5	GND	V_{PP}	"L"	Data input	Address input	"L"	GND
Verification	+5	GND	"L"	"L"	Data output	Address input	"L"	GND
Inhibition of programming/verification	+5	GND	Don't care	"H"	High impedance	Don't care	"L"	GND

"H"; V_{IH} level, "L": V_{IL} level

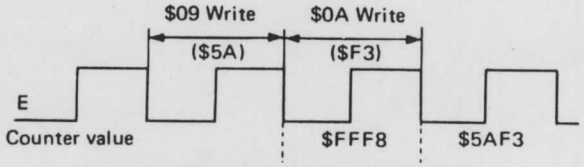
- Control/Status Register 1 (8 bit)
- Control/Status Register 2 (7 bit)
- Free Running Counter (16 bit)
- Output Compare Register 1 (16 bit)
- Output Compare Register 2 (16 bit)
- Input Capture Register (16 bit)

● Free-Running Counter (FRC) ($0009 : 000A)

The key timer element is a 16-bit Free-Running Counter which is incremented by system clock (E). The counter value is readable by software without affecting the FRC. It is cleared by reset.

A write to the high byte of the FRC ($09) will preset the high and low byte of the FRC to $FFF8 . A continuous write to the high and low byte FRC, however, will set them to the write data.

The FRC write timing will be as follows when double store instructions (STD, STX etc.) execute.



In the case of a write ($5AF3) to the FRC.

Figure 21 Counter Write Timing

● Output Compare Register (OCR) ($000B, $000C; OCR1) ($0019, $001A; OCR2)

The Output Compare Register is a 16-bit read/write register used to control an output waveform. It is always compared with the FRC on each E-cycle.

When a match is found, Output Compare Flag (OCF) in the Timer Control/Status Register (TCSR) is set. If an output enable bit (OE) in the TCSR2 is "1", an output level bit (OLVL) in the TCSR will appear at Port 21 (Tout 1) or Port 25 (Tout 2).

The OCR and OLVL can then be changed for the next compare. The OCR is set to $FFFF at reset. The compare function is inhibited for a cycle after a write to the OCR or to the high byte of the FRC. This is to set the 16-bit value valid in the register for compare. In addition, it is because $FFF8 is set at the next cycle of a write to the high byte of the FRC.

* For a write to the FRC or the OCR, 2-byte transfer instruction (such as STX etc.) should be used.

● Input Capture Register (ICR) ($000D : 000E)

The Input Capture Register is a 16-bit read only register used to store the FRC when an external input transition occures defined by input edge bit (IEDG) in the TCSR1.

In order to input the external signal to the edge detective circuit, Port 20 should be configured as an input. When an input capture occures at the next cycle of a read the high-byte of the ICR, the input capture will delay one cycle. In order to ensure the input capture, a read to the ICR needs 2-byte transfer instruction, and the input pulse width should be at least 2 system cycles. This register is cleared ($0000) at reset.

● Timer Control/Status Register 1 (TCSR1) ($0008)

The Timer Control/Status Register 1 is an 8-bit register of which all bits are readable while the lower 5 bits can be written. The upper 3 bits indicate the following timer's status.

Bit 5 The FCR has overflowed. (TOF).

Bit 6 A match has been found between the FCR and the OCR 1 (OCF1).

Bit 7 A level transition of the timer input has been detected (ICF).

The followings are each bit descriptions.

Timer Control/Status Register 1

7	6	5	4	3	2	1	0	
ICF	OCF1	TOF	EICI	EOCI1	ETOI	IEDG	OLVL1	$0008

Bit 0 OLVL1 Output Level 1

When a match is found between the FCR and the OCR1, OLVL1 will appear at Port 21 if OE1, bit 0 of the TCSR2, is set.

Bit 1 IEDG Input Edge

This bit controls which level transition will trigger the FCR transfer to the ICR. For this function, the DDR corresponding to Port 20 should be cleared.

IEDG=0, transferred on a negative edge

IEDG=1, transferred on a positive edge

Bit 2 ETOI Enable Timer Overflow Interrupt

When this bit is set, an internal interrupt (IRQ_3) is enabled for TOI. When cleared, the interrupt is inhibited.

Bit 3 EOCI1 Enable Output Compare Interrupt 1

When this bit is set, an internal interrupt (IRQ_3) is enabled for OCI1. When cleared, the interrupt is inhibited.

Bit 4 EICI Enable Input Capture Interrupt

When this bit is set, an internal interrupt (IRQ_3) is enabled for ICI. When cleared, the interrupt is inhibited.

Bit 5 TOF Timer Overflow Flag

This read only bit is set when the FCR contains all 1's. It is cleared by reading the TCSR1 followed by the FCR's high byte ($0009).

Bit 6 OCF1 Output Compare Flag 1

This read only bit is set when a match is found between the OCR1 and the FRC. It is cleared by writing to the OCR1 ($000B or $000C) followed by reading the TCSR1 or TCSR2.

Bit 7 ICF Input Capture Flag

This read only bit is set to indicate a level transition defined by IEDG. It is cleared by reading the high byte ($000D) of the ICR followed by the TCSR1 or TCSR2.

● Timer Control/Status Register 2 (TCSR2) ($000F)

The Timer Control/Status Register 2 is a 7-bit register. All bits are readable while the lower 4 bits can be written. The upper 3 bits indicate the following timer's status.

Bit 5 A match has been found between the FRC and the OCR2 (OCF2).

Bit 6 The same flag as the OCF1 of the TCSR1.

Bit 7 The same flag as the ICF of the TCSR1.

The followings are each bit descriptions.

Timer Control/Status Register 2

7	6	5	4	3	2	1	0	
ICF	OCF1	OCF2	–	EOCI2	OLVL2	OE2	OE1	$000F

Bit 0 OE1 Output Enable 1

If this bit is set, the OLVL1 will appear at Port 21 when a match is found between the FCR and the OCR1. When it is cleared, Port 21 will be I/O port. When set, it will be an output of OLVL1 automatically.

Bit 1 OE2 Output Enable 2

If this bit is set, the OLVL2 will appear at Port 25 when a match between the FCR and the OCR2. When this bit is cleared, Port 25 will be I/O port. When set, it will be an output of OLVL2 automatically.

Bit 2 OLVL2 Output Level 2

OLVL2 is transferred to Port 25 when a match is found between the FCR and the OCR2. If OE2, bit 5 of the TCSR2, is set, OLVL2 will appear at Port 25.

Bit 3 EOCI2 Enable Output Compare Interrupt 2

When this bit is set, an internal interrupt (IRQ_3) is enabled for OCI2. When cleared, the interrupt is inhibited.

Bit 4 Not Used

Bit 5 OCF2 Output Compare Flag 2

This read-only bit is set when a match is found between the FCR and the OCR2. It is cleared by writing to the OCR2 ($0019 or $001A) followed by reading the TCSR2.

Bit 6 OCF1 Output Compare Flag 1

Bit 7 ICF Input Capture Flag

OCF1 and ICF addresses are partially decoded. CPU read of the TCSR1/TCSR2 makes it possible to read OCF1 and ICF into bit 6 and bit 7.

Both the TCSR1 and TCSR2 will be cleared by reset.

(Note) If OE1 or OE2 is set before the first output compare match is found after reset, Port 21 and Port 25 will output "0" respectively.

(Note) Because the set condition of ICF precedes its reset condition, ICF is not cleared when the set condition and the reset condition occur simultaneously. The same phenomenon applies to OCF1, OCF2 or TOF respectively.

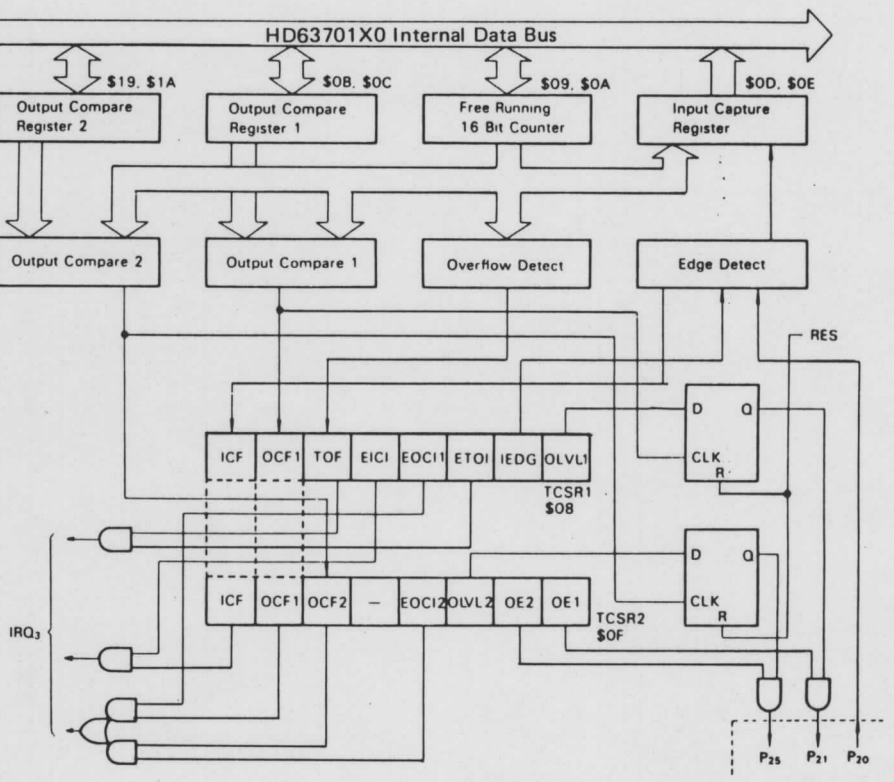


Figure 22 Timer 1 Block Diagram

■ TIMER 2

In addition to the timer 1, the HD63701X0 provides an 8-bit reloadable timer, which is capable of counting the external event. This timer 2 contains a timer output, so the MCU can generate three independent waveforms. (refer to Fig. 23.)

The timer 2 is configured as follows:

Control/Status Register 3 (7 bit)
8-bit Up Counter
Time Constant Register (8 bit)

● Timer 2 Up Counter (T2CNT) ($001D)

This is an 8-bit up counter which is incremented by the clock controlled by CKS0 and CKS1 of the TCSR3. The T2CNT is always readable without affecting itself. In addition, any value can be written to the T2CNT by software even during counting.

The counter is cleared when a match is found between the T2CNT and the TCONR or by reset.

A write to the T2CNT at the clear cycle does not reset it but put the data to it.

● Time Constant Register (TCONR) ($001C)

The Timer Constant Register is an 8-bit write only register. It is always compared with the T2CNT.

When a match has been found, counter match flag (CMF) of the Timer Control/Status Register 3 (TCSR3) is set and the value selected by TOS0 and TOS1 of the TCSR3 will appear at Port 26. When CMF is set, the FCR will be cleared simultaneously and then a counting starts from $00. This enables regular interrupts and waveform outputs without any software support. The TCONR is set to "$FF" by reset.

● Timer Control/Status Register 3 (TCSR3) ($001B)

The Timer Control/Status Register 3 is a 7-bit register. All bits are readable while 6 bits except for CMF can be written.

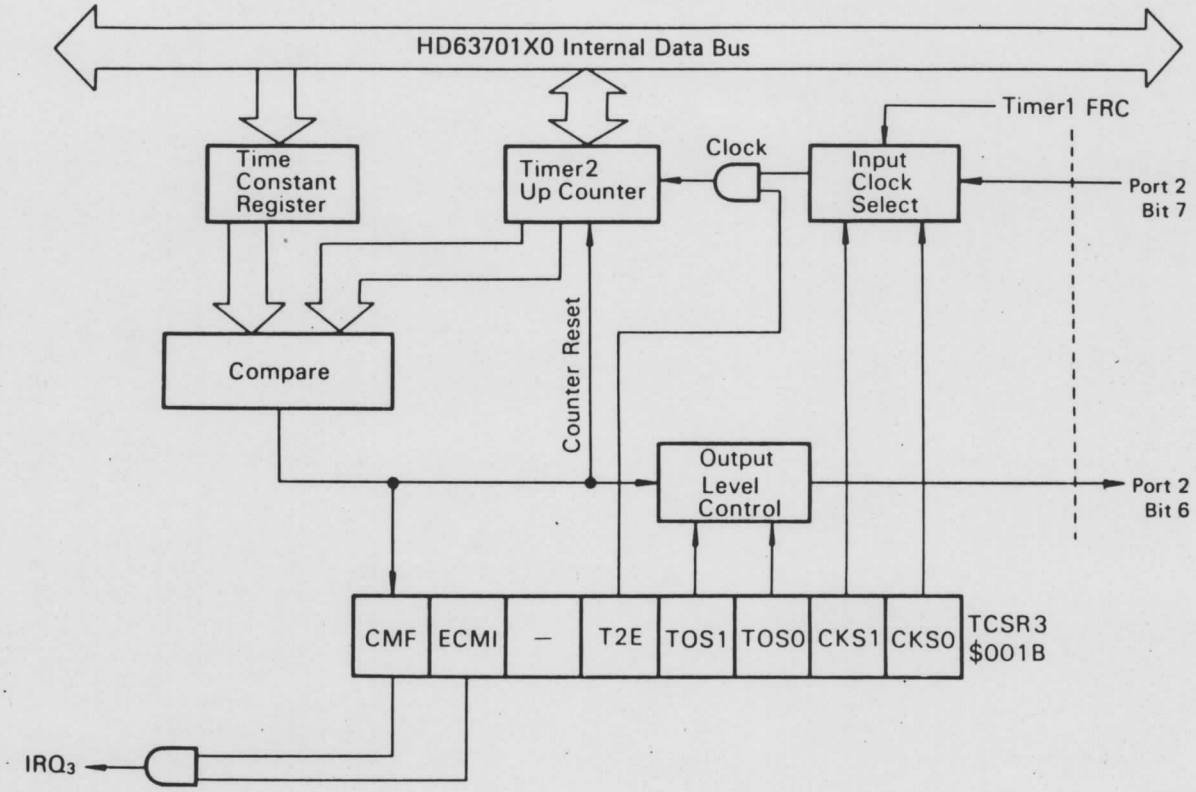


Figure 23 Timer 2 Block Diagram

The followings are each bit descriptions.

Timer Control/Status Register 3

7	6	5	4	3	2	1	0	
CMF	ECMI	–	T2E	TOS1	TOS0	CKS1	CKS0	$001B

Bit 0 CKS0 Input Clock Select 0
Bit 1 CKS1 Input Clock Select 1

An input clock to the T2CNT is selected by these bits as shown in Table 7. When an external clock is selected, Port 27 will be an input automatically. The positive edge of the external clock increments the T2CNT. The maximum external clock is half of the system clock frequency.

Table 7 Input Clock Select

CKS1	CKS0	Input Clock to the Counter
0	0	E clock
0	1	E clock/8*
1	0	E clock/128*
1	1	External clock

* These clocks come from the FRC of the timer 1. If one of these clocks is selected as an input clock to the up counter, a write to the FRC of the timer 1 should be inhibited.

Bit 2 TOS0 Timer Output Select 0
Bit 3 TOS1 Timer Output Select 1

When a match is found between the T2CNT and the TCONR, timer 2 output selected by these bits shown in Table 8 will appear at Port 26. When both TOS0 and TOS1 are cleared, Port 26 will be an I/O port.

Table 8 Timer 2 Output Select

TOS1	TOS0	Timer Output
0	0	Timer Output Inhibited
0	1	Toggle Output*
1	0	Output "0"
1	1	Output "1"

* When a match is found between the T2CNT and the TCONR, timer 2 output level is reversed. This leads to production of a square wave with 50% duty to the external without any software support.

Bit 4 T2E Timer 2 Enable Bit

When this bit is cleared, the T2CNT will stop. When set, a clock selected by CKS1 and CKS0 (Table 7) provides to the T2CNT.

(Note) P_{26} is "0" when T2E is cleared and P_{26} is configured as an output by TOS1 or TOS0. It also is "0" when T2E is set and P_{26} is configured as an output before the first counter match.

Bit 5 Not Used
Bit 6 ECMI Enable Counter Match Interrupt

When this bit is set, an internal interrupt (IRQ_3) is enabled for CMI. When cleared, the interrupt is inhibited.

Bit 7 CMF Counter Match Flag

This read only bit is set when a match is found between the T2CNT and the TCONR. It is cleared by writing "0". (It cannot be written "1" by software).

Each bit of the TCSR3 is cleared by reset.

■ SERIAL COMMUNICATION INTERFACE (SCI)

The HD63701X0 SCI provides two operation modes; one is an asynchronous mode by the NRZ format and the other is a clocked synchronous mode to transfer data synchronizing with the serial clock.

The serial interface is configured as follows:

- Transmit/Receive Control and Status Register (TRCSR)
- Rate/Mode Control Register (RMCR)
- Receive Data Register (RDR)
- Receive Data Shift Register (RDSR)
- Transmit Data Register (TDR)
- Transmit Data Shift Register (TDSR)

The SCI is initialized by software. The procedure is usually as follows:

1) Write a operation mode into each corresponding control bit of the RMCR.
2) Write a operation mode into each corresponding control bit of the TRCSR.

When setting the baud rate and operation mode, TE and RE should be "0". When TE and RE is set again, more than 1 bit cycle of the current baud rate is necessary. If set in less than 1 bit cycle, the SCI cannot be initialized occasionally.

● Asynchronous Mode

An asynchronous mode contains the following two data formats:

1 Start Bit + 8 Bit Data + 1 Stop Bit; 8 Bit Data Format
1 Start Bit + 9 Bit Data + 1 Stop Bit; 9 Bit Data Format

In 9 Bit Data Format, if the 9th bit is "1", the format of
1 Start Bit + 8 Bit Data + 2 Stop Bit

The SCI is initialized by writing desirable control bytes to the RMCR and then to the TRCSR.

The transmit operation is enabled by TE in the TRCSR. When TE is set, the output of the TDSR is connected to P_{24} which will be configured as an output regardless of the DDR, and then the serial output is initiated by transmitting to a 10-bit preamble of "1" in the 8 Bit Data Format or an 11-bit preamble of "1" in the 9 Bit Data Format. Following the preamble, the internal synchronization is established and the transmitter section is ready for operation.

At this point one of two situation exist:

1) If the TDR is empty (TDRE=1), a continuous string of ones will be sent indicating an idle line.
2) If a byte has been written to the TDR (TDRE=0), it is transferred to the TDSR, TDRE will be set and transmission will begin.

During the transfer itself, the start bit (0) is first transmitted. Then the 8 data bits or the 9 data bits (beginning with bit 0) followed by the stop bit (1) are transmitted. When the TDR has been emptied, TDRE is set.

If the MCU fails to respond to the flag within the proper time, (TDRE is still set when the next normal transfer from the TDR to the TDSR should occure) then a "1" will be sent (instead of a "0") at start bit time, followed by more 1's until more data is supplied to the TDR. No 0's will be sent while TDRE remains as "1".

The receive operation is enabled by RE which configures P_{23}. The receive operation is controlled by the contents of the TRCSR and the RMCR. The receiver bit interval is divided into 8 sub-intervals for internal synchronization. The received bit stream is synchronized by the first "0" (space) encountered. The approximate center of each bit time is strobed during the next 10 bits.

If the tenth bit is not a "1" (stop bit), a framing error is assumed and ORFE is set. If the tenth bit is a "1", the data is transferred to the RDR and interrupt flag RDRF is set. If RDRF is still set at the next tenth bit time, ORFE will be set, indicating an over-run has occurred. When the CPU responds to either flag

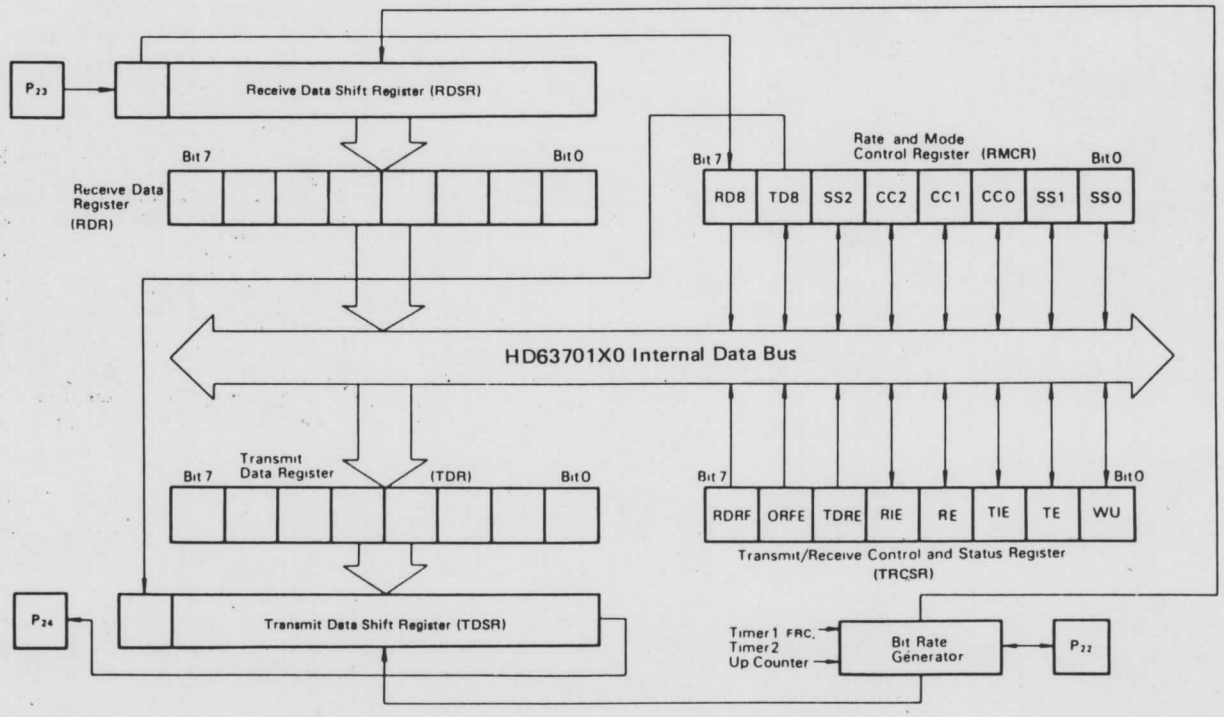


Figure 24 Serial Communication Interface Block Diagram

(RDRF or ORFE) by reading the TRCSR followed by reading the RDR, RDRF (or ORFE) will be cleared.

(Note) Clock Source in Asynchronous Mode

When using an internal clock for the SCI, the following requirements are applicable:

- Set CC1 and CC0 to "1" and "0" respectively.
- A clock is generated regardless of the value of TE, RE.
- The maximum clock rate is E÷16.
- The output clock is the same as the bit rate.

When using an external clock for the SCI, the following requirements are applicable:

- Set CC1 and CC0 in the RMCR to "1" and "1" respectively.
- The external clock should be set 16 times the desired baud rate.
- the maximum clock frequency is the same as the system clock.

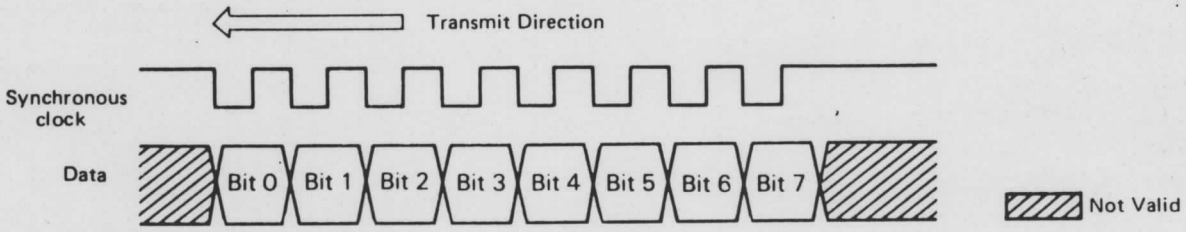
- **Clocked Synchronous Mode**

In the clocked synchronous mode, the transmit operation is synchronized with the clock pulse. In the clocked synchronous mode an SCI clock I/O pin is only P_{22}, so the receive and transmit operation cannot be simultaneously enabled. Therefore, TE and RE should not be set simultaneously. Fig. 25 gives a synchronous clock and a data format in the clocked synchronous mode.

The transmit operation is enabled by TE in the TRCSR. P_{24} is configured as an output regardless of the value of the corresponding DDR.

Both the RMCR and TRCSR should be set in the desirable operating conditions for data transmit.

If the user wishes to provide an external clock, the data bits (beginning with bit 0) are transmitted from P_{24}, synchronizing with 8 clock pulses supplied to P_{22}, when TDRE is "0". TDRE is set when the TDSR is "empty". More the 9th clock pulse is ignored.



- Transmit data is sent between the negative edge of a synchronous clock and the next negative edge.
- Receive data is latched at the positive edge.

Figure 25 Clocked Synchronous Mode Format

The receive operation is enabled by RE. P_{22} is configured as an input for the 8 bit external clock and P_{23} is configured as an input for the receive data. The operating mode of data receive is decided by the TRCSR and the RMCR.

If the external clock is provided, RE should be set when P_{22} is "High". The receive data is transferred to the RDSR by this clock, and RDRF is set. More the 9th clock pulse are ignored. When RDRF is cleared by reading the RDR, the MCU starts receiving the next data.

RDRF, therefore, should be cleared with P_{22} "High". When the first byte data is received, RDRF is set. After the second byte, the receive operation is enabled by clearing RDRF.

- **Transmit/Receive Control and Status Register (TRCSR) ($0011)**

The TRCSR is an 8 bit register which is readable. Bits 0 to 4 are also writable. This register is initialized to $20 by reset. Each bit functions as follows.

Transmit/Receive Control Status Register

7	6	5	4	3	2	1	0	
RDRF	ORFE	TDRE	RIE	RE	TIE	TE	WU	$0011

Bit 0 WU Wake-up

In a typical multi-processor configuration, the software protocol will usually identify the address at the beginning of the message. In order to permit uninterested MCU's to ignore the remaining message, a wake-up function is available. By this, uninterested MCU can inhibit all further receive processing till the next message starts.

Then wake-up function is triggered by consecutive 1's with 1 frame length (10 bits for the 8-bit data format, or 11 bits for the 9-bit data format). The software protocol should provide the idle time between messages.

By setting this bit, the MCU stops data receive till the next message. The receive of consecutive "1" with one frame length wakes up and clears this bit and then the MCU restarts the receive operation. However, the RE flag should be set before setting this bit. In the clocked synchronous mode WU is not available, so this bit should not be set.

Bit 1 TE Transmit Enable

When this bit is set, transmit data will appear at P_{24} after one frame preamble in asynchronous mode, while in clocked synchronous mode appear immediately. This is executed regardless of the value of the corresponding DDR. When TE is cleared, the serial I/O doesn't affect P_{24}.

Bit 2 TIE Transmit Interrupt Enable

When this bit is set, an internal interrupt (IRQ_3) is enabled when TDRE (bit 5) is set. When cleared, the interrupt is inhibited.

Bit 3 RE Receive Enable

When set, P_{23} is configured as an input for the receive operation regardless of the value of the DDR. When RE is cleared, the serial I/O doesn't affect P_{23}.

Bit 4 RIE Receive Interrupt Enable

When this bit is set, an internal interrupt, IRQ_3 is enabled when RDRF (bit 7) or ORFE (bit 6) is set. When cleared, the interrupt is inhibited.

Bit 5 TDRE Transmit Data Register Empty

TDRE is set when the TDR is transferred to the TDSR in the asynchronous mode, while it is set when the TDSR is "empty" in clocked synchronous mode. This bit is cleared by reading the TRCSR and writing the new transmit data to the TDR. TDRE is set by reset.

(Note) TE should be set before clearing TDRE.

Bit 6 ORFE Overrun Framing Error

ORFE is set when an overrun or a framing error is occured (during data receive only). An overrun error occurs when a new receive data is ready to be transferred to the RDR with RDRF still set. A framing error occurs when a stop bit is "0". But in clocked synchronous mode, this bit is not affected. This bit is cleared when reading the TRCSR, then the RDR, or by reset.

Bit 7 RDRF Receive Data Register Full

RDRF is set when the RDSR is transferred to the RDR. Cleared when reading the TRCSR, then the RDR, or by reset.

(Note) When a few bits are set between bit 5 to bit 7 in the TRCSR, a read of the TRCSR is sufficient for clearing those bits. It is not necessary to read the TRCSR everytime to clear each bit.

- **Rate/Mode Control Register (RMCR)**

The RMCR controls the followings:

- Baud Rate
- Clock Source
- Data Format
- P_{22} Function

In addition, if 9-bit data format is set in the asynchronous mode, the 9th bit is put in this register. All bits are readable and writable except bit 7 (read only). This register is cleared by reset.

These bits select the baud rate when using the internal clock. Table 9 lists the available bit times and baud rates. The timer 1's FRC (SS2=0) and the timer 2's up counter (SS2=1) provide

Rate/Mode Control Register

7	6	5	4	3	2	1	0	
RD8	TD8	SS2	CC2	CC1	CC0	SS1	SS0	$0010

Bit 0 SS0
Bit 1 SS1 } Speed Select
Bit 5 SS2

the internal clock for the SCI. When the source of the SCI internal clock is the timer 2's up counter, the desired baud rates may be selected by the TCONR shown in Table 10.

(Note) When operating the SCI with internal clock, do not write to the counter which is the source of the SCI clock.

Bit 2 CC0
Bit 3 CC1 } Clock Control/Format Select*
Bit 4 CC2

These bits select the data format and the clock source (refer to Table 11).

* CC0, CC1 and CC2 are cleared and the MCU will be in the clocked synchronous mode (the external clock operation) by reset. Then P_{22} is forced to be configured as an input for the clock. If using P_{22} for an output, the DDR of port 2 should be set to "1" and CC1, CC0 must be set to "01".

Table 9 SCI Bit Times and Rates

(1) Asynchronous Mode

			XTAL	2.4576MHz	4.0MHz	4.9152MHz
SS2	SS1	SS0	E	614.4kHz	1.0MHz	1.2288MHz
0	0	0	E÷16	26μs/38400Baud	16μs/62500Baud	13μs/76800Baud
0	0	1	E÷128	208μs/4800Baud	128μs/7812.5Baud	104.2μs/9600Baud
0	1	0	E÷1024	1.67ms/600Baud	1.024ms/976.6Baud	833.3μs/1200Baud
0	1	1	E÷4096	6.67ms/150Baud	4.096ms/244.1Baud	3.333ms/300Baud
1	—	—	—	*	*	*

* When SS2 is "1", Timer 2 provides SCI clocks. The baud rate is shown as follows with the TCONR as N.

$$\text{Baud Rate} = \frac{f}{32\,(N+1)} \qquad \left(\begin{array}{l} f: \text{input clock frequency to the timer 2 counter} \\ N = 0 \sim 255 \end{array}\right)$$

(2) Clocked Synchronous Mode *

			XTAL	4.0MHz	6.0MHz	8.0MHz
SS2	SS1	SS0	E	1.0MHz	1.5MHz	2.0MHz
0	0	0	E÷2	2μs/bit	1.33μs/bit	1μs/bit
0	0	1	E÷16	16μs/bit	10.7μs/bit	8μs/bit
0	1	0	E÷128	128μs/bit	85.3μs/bit	64μs/bit
0	1	1	E÷512	512μs/bit	341μs/bit	256μs/bit
1	—	—	—	**	**	**

* Bit rates in the case of internal clock operation. In the case of external clock operation, the external clock is operatable up to DC ~ 1/2 system clock.

** The bit rate is shown as follows with the TCONR as N.

$$\text{Bit Rate } (\mu s/\text{bit}) = \frac{4\,(N+1)}{f} \qquad \left(\begin{array}{l} f: \text{input clock frequency to the timer 2 counter} \\ N = 0 \sim 255 \end{array}\right)$$

Table 10 Baud Rate and Time Constant Register Example

Baud Rate (Baud) \ XTAL	2.4576MHz	3.6864MHz	4.0MHz	4.9152MHz	8.0MHz
110	21*	32*	35*	43*	70*
150	127	191	207	255	51*
300	63	95	103	127	207
600	31	47	51	63	103
1200	15	23	25	31	51
2400	7	11	12	15	25
4800	3	5	–	7	12
9600	1	2	–	3	–
19200	0	–	–	1	–
38400	–	–	–	0	–

*E/8 clock is provided to the timer 2's up counter.

Table 11 SCI Format and Clock Source Control

CC2	CC1	CC0	Format	Mode	Clock Source	Port 2, Bit 2	Port 2, Bit 3	Port 2, Bit 4
0	0	0	8-bit data	Clocked Synchronous	External	Input	When RE is "1", bit 3 is used for a serial input.	When TE is "1", bit 4 is used for a serial output.
0	0	1	8-bit data	Asynchronous	Internal	Not Used**		
0	1	0	8-bit data	Asynchronous	Internal	Output*		
0	1	1	8-bit data	Asynchronous	External	Input		
1	0	0	8-bit data	Clocked Synchronous	Internal	Output		
1	0	1	9-bit data	Asynchronous	Internal	Not Used**		
1	1	0	9-bit data	Asynchronous	Internal	Output*		
1	1	1	9-bit data	Asynchronous	External	Input		

* Clock output regardless of RE or TE in the TRCSR.
** Not used for the SCI.

Bit 6 TD8 Transmit Data Bit 8
When selecting the 9-bit data format in the asynchronous mode, this bit is transmitted as the 9th data.

Bit 7 RD8 Receive Data Bit 8
When selecting the 9-bit data format in the asynchronous mode, this bit stores the 9th bit data.

■ TIMER, SCI STATUS FLAG

Table 12 shows set and clear conditions of each status flag in the timer 1, the timer 2 and the SCI.

If the flag set and clear conditions occur at the same time, the flag of the Timer 1 and the Timer 2 will be set, and the SCI cleared. Therefore the OCF1 and OCF2 of the Timer 1 may not be cleared correctly because set signal is generated periodically whenever the OCR matches the FRC. In order to clear these flags correctly, the match should be prohibited during the period between reading the TSCR and writing the OCR. For instance, these flags will be cleared correctly if the TCSR is read and the OCR is written continuously soon after matching the value of the OCR and the FCR.

Refer to "APPLICATION NOTES—Cautions for OCF clearing in the TCSR of the Timer 1" for detail.

Table 12 Timer 1, Timer 2 and SCI Status Flag

		Set Condition	Reset Condition
Timer 1	ICF	FRC → ICR by edge input to P_{20}.	1. Read the TCSR1 or TCSR2 then ICRH, when ICF=1 2. $\overline{RES}$=0
	OCF1	OCR1=FRC	1. Read the TCSR1 or TCSR2 then write to the OCR1H or OCR1L, when OCF1=1 2. $\overline{RES}$=0
	OCF2	OCR2=FRC	1. Read the TCSR2 then write to the OCR2H or OCR2L, when OCF2=1 2. $\overline{RES}$=0
	TOF	FRC=$FFFF+1 cycle	1. Read the TCSR1 then FRCH, when TOF=1 2. $\overline{RES}$=0
Timer 2	CMF	T2CNT=TCONR	1. Write "0" to CMF, when CMF=1 2. $\overline{RES}$=0
SCI	RDRF	Receive Shift Register → RDR	1. Read the TRCSR then RDR, when RDRF=1 2. $\overline{RES}$=0
	ORFE	1. Framing Error (Asynchronous Mode) Stop Bit = 0 2. Overrun Error (Asynchronous Mode) Receive Shift Register → RDR when RDRF=1	1. Read the TRCSR then RDR, when ORFE=1 2. $\overline{RES}$=0
	TDRE	1. Asynchronous Mode TDR → Transmit Shift Register 2. Clocked Synchronous Mode Transmit Shift Register is "empty" 3. $\overline{RES}$=0	Read the TRCSR then write to the TDR, when TDRE=1 (Note) Clear TDRE after setting TE.

(Note) 1. →; transfer
2. For example; "ICRH" means High byte of ICR.

■ LOW POWER DISSIPATION MODE

The HD63701X0 provides two low power dissipation modes; sleep and standby.

● Sleep Mode

The MCU will be in the sleep mode when SLP instruction is executed. In the sleep mode, the CPU stops and the registers' contents are retained. While the peripherals such as timers, SCI etc. continue their functions. The power dissipation of the sleep-condition is one fifth that of the operating condition.

The MCU returns from this mode by an interrupt, $\overline{RES}$ or $\overline{STBY}$; it will be reset by $\overline{RES}$ and the standby mode by $\overline{STBY}$. When the CPU responds to an interrupt request, it cancels the sleep mode, returns to the operation mode and branches to the interrupt routine. When the CPU masks this interrupt, it cancels the sleep mode and executes the next instruction. However, for example if the timer 1 or 2 prohibits a timer interrupt, the CPU doesn't cancel the sleep mode because of no interrupt request.

This sleep mode is effective to reduce the power dissipation for a system with no need of the HD63701X0's consecutive operation.

● Standby Mode

In MCU mode, the HD63701X0 stops and reset with $\overline{STBY}$ "low". In this mode, the power dissipation is reduced conspicuously. All pins except for the power supply, $\overline{STBY}$ and XTAL are detached from the MCU internally and will be the high impedance state.

While the contents of RAM is retained. The MCU returns from this mode by reset. The followings are typical usage of this mode.

Save the CPU information and SP contents on RAM by $\overline{NMI}$. Then disable the RAME bit of the RAM control register and set the STBY PWR bit to go to the standby mode. If the STBY PWR bit is still set at reset, that indicates the power is supplied to the MCU and RAM contents are retained properly. So system can restore itself by returning their pre-standby informations to the SP and the CPU. Fig. 26 depicts the timing at each pin with this example.

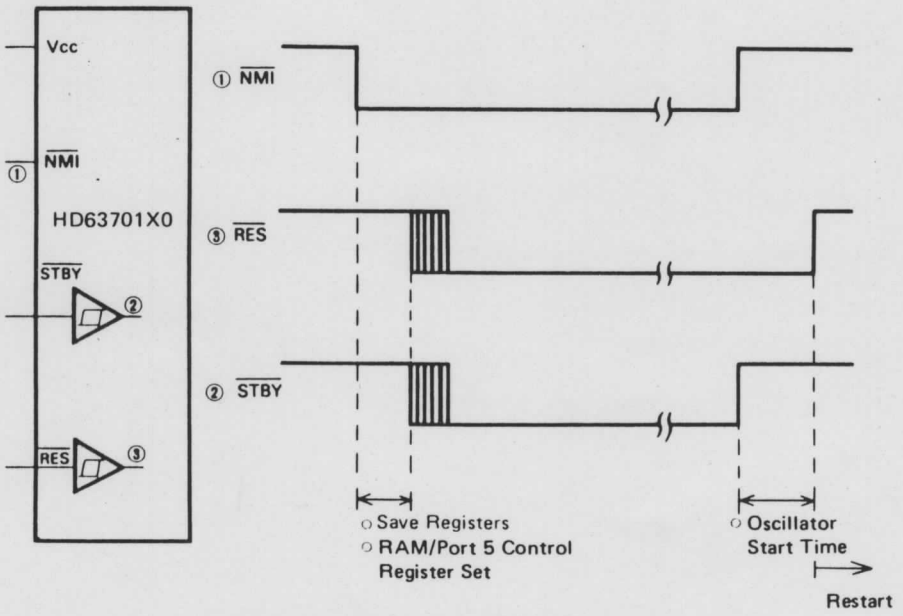


Figure 26 Standby Mode Timing

■ TRAP FUNCTION

The CPU generates an interrupt with the highest priority (TRAP) when fetching an undefined instruction or an instruction from non-memory space. The TRAP prevents the system-burst caused by noise or a program error.

● Op Code Error

When fetching an undefined op code, the CPU saves CPU registers as well as a normal interrupt and branches to the TRAP ($FFEE, $FFEF). This provides the priority next to reset.

● Address Error

When an instruction fetch is made excluding internal PROM, RAM and external memory area, the MCU generates an interrupt as well as an op code error. But on the system with no memory in its external memory area, this error processing is not applicable if an instruction fetch is made from the external non-memory area. Table 13 provides addresses where an address error occurs to each mode.

This processing is available only for an instruction fetch and is not applicable to the access of normal data read/write.

Table 13 Addresses Applicable to Address Errors

Mode	1	2	3
Address	$0000 ≀ $001F	$0000 ≀ $001F	$0000 ≀ $003F $0100 ≀ $EFFF

(Note) The TRAP interrupt provides a retry function differently from other interrupts. This is a program flow return to the address where the TRAP occurs when a sequence returns to a main routine from the TRAP interrupt routine by RTI. The retry can prevent the system burst caused by noise etc.
However, if another TRAP occurs, the program repeats the TRAP interrupt forever, so the consideration is necessary in programming.

■ INSTRUCTION SET

The HD63701X0 provides object code upward compatible with the HD6801 to utilize all instruction set of the HMCS6800. It also reduces the execution times of key instructions for throughout improvement.

Bit manipulation instruction, change instruction of the index register and accumulator and sleep instruction are also added.

The followings are explained here.

- CPU Programming Model (refer to Fig. 27)
- Addressing Mode
- Accumulator and Memory Manipulation Instruction (refer to Table 14)
- New Instruction
- Index Register and Stack Manipulation Instruction (refer to Table 15)
- Jump and Branch Instruction (refer to Table 16)
- Condition Code Register Manipulation (refer to Table 17)
- Op Code Map (refer to Table 18)

● Programming Model

Fig. 27 depicts the HD63701X0 programming model. The double accumulator D consists of accumulator A and B, so when using the accumulator D, the contents of A and B are destroyed.

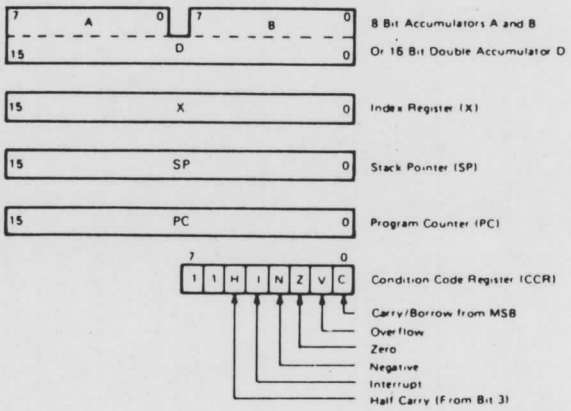


Figure 27 CPU Programming Model

● CPU Addressing Mode

The HD63701X0 provides 7 addressing modes. The addressing mode is decided by an instruction type and code. Table 14 through 18 show addressing modes of each instruction with the execution times counted by the machine cycle.

When the clock frequency is 4MHz, the machine cycle time becomes microseconds directly.

Accumulator (ACCX) Addressing

Only an accumulator is addressed and the accumulator A or B is selected. This is a one-byte instruction.

Immediate Addressing

This addressing locates a data in the second byte of an instruction. However, LDS and LDX locate a data in the second and third byte exceptionally. This addressing is a 2 or 3-byte instruction.

Direct Addressing

In this addressing mode, the second byte of an instruction shows the address where a data is stored. 256 bytes ($0 through $255) can be addressed directly. Execution times can be reduced by storing data in this area so it is recommended to make it RAM for users' data area in configurating a system. This is a 2-byte instruction, while 3 byte with regard to AIM, OIM, EIM and TIM.

Extended Addressing

In this mode, the second byte shows the upper 8 bit of the data stored address and the third byte the lower 8 bit. This indicates the absolute address of 3 byte instruction in the memory.

Indexed Addressing

The second byte of an instruction and the lower 8 bit of the index register are added in this mode. As for AIM, OIM, EIM and TIM, the third byte of an instruction and the lower 8 bits of the index register are added.

This carry is added to the upper 8 bit of the index register and the result is used for addressing the memory. The modified address is retained in the temporary address register, so the contents of the index register doesn't change. This is a 2-byte instruction except AIM, OIM, EIM and TIM (3-byte instruction).

Implied Addressing

An instruction itself specifies the address. That is, the instruction addresses a stack pointer, index register etc. This is a one-byte instruction.

Relative Addressing

The second byte of an instruction and the lower 8 bits of the program counter are added. The carry or borrow is added to the upper 8 bit. So addressing from −126 to +129 byte of the current instruction is enabled. This is a 2-byte instruction.

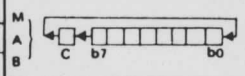
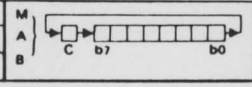
(Note) CLI, SEI Instructions and Interrupt Operation

When accepting the IRQ at a preset timing with the help of CLI and SEI instructions, more than 2 cycles are necessary between the CLI and SEI instructions. For example, the following program (a) (b) don't accept the IRQ but (c) accepts it.

·	·	·
·	·	·
·	·	·
·	·	CLI
CLI	CLI	NOP
SEI	NOP	NOP
·	SEI	SEI
·	·	·
·	·	·
·	·	·
·	·	·
(a)	(b)	(c)

The same thing can be said to the TAP instruction instead of the CLI and SEI instructions.

Table 14 Accumulator, Memory Manipulation Instructions

Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3				A + M → A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				B + M → B	↕	•	↕	↕	↕	↕
Add Double	ADDD	C3	3	3	D3	4	2	E3	5	2	F3	5	3				A B + M M + 1 → A B	•	•	↕	↕	↕	↕
Add Accumulators	ABA													1B	1	1	A + B → A	↕	•	↕	↕	↕	↕
Add With Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				B + M + C → B	↕	•	↕	↕	↕	↕
AND	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				A·M → A	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3				B·M → B	•	•	↕	↕	R	•
Bit Test	BIT A	85	2	2	95	3	2	A5	4	2	B5	4	3				A·M	•	•	↕	↕	R	•
	BIT B	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B·M	•	•	↕	↕	R	•
Clear	CLR							6F	5	2	7F	5	3				00 → M	•	•	R	S	R	R
	CLRA													4F	1	1	00 → A	•	•	R	S	R	R
	CLRB													5F	1	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A - M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B - M	•	•	↕	↕	↕	↕
Compare Accumulators	CBA													11	1	1	A - B	•	•	↕	↕	↕	↕
Complement, 1's	COM							63	6	2	73	6	3				$\overline{M}$ → M	•	•	↕	↕	R	S
	COMA													43	1	1	$\overline{A}$ → A	•	•	↕	↕	R	S
	COMB													53	1	1	$\overline{B}$ → B	•	•	↕	↕	R	S
Complement, 2's (Negate)	NEG							60	6	2	70	6	3				00 - M → M	•	•	↕	↕	①	②
	NEGA													40	1	1	00 - A → A	•	•	↕	↕	①	②
	NEGB													50	1	1	00 - B → B	•	•	↕	↕	①	②
Decimal Adjust, A	DAA													19	2	1	Converts binary add of BCD characters into BCD format	•	•	↕	↕	↕	③
Decrement	DEC							6A	6	2	7A	6	3				M - 1 → M	•	•	↕	↕	④	•
	DECA													4A	1	1	A - 1 → A	•	•	↕	↕	④	•
	DECB													5A	1	1	B - 1 → B	•	•	↕	↕	④	•
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				A ⊕ M → A	•	•	↕	↕	R	•
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				B ⊕ M → B	•	•	↕	↕	R	•
Increment	INC							6C	6	2	7C	6	3				M + 1 → M	•	•	↕	↕	⑤	•
	INCA													4C	1	1	A + 1 → A	•	•	↕	↕	⑤	•
	INCB													5C	1	1	B + 1 → B	•	•	↕	↕	⑤	•
Load Accumulator	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3				M → A	•	•	↕	↕	R	•
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3				M → B	•	•	↕	↕	R	•
Load Double Accumulator	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3				M + 1 → B, M → A	•	•	↕	↕	R	•
Multiply Unsigned	MUL													3D	7	1	A x B → A : B	•	•	•	•	•	⑪
OR, Inclusive	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3				A + M → A	•	•	↕	↕	R	•
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3				B + M → B	•	•	↕	↕	R	•
Push Data	PSHA													36	4	1	A → Msp, SP - 1 → SP	•	•	•	•	•	•
	PSHB													37	4	1	B → Msp, SP - 1 → SP	•	•	•	•	•	•
Pull Data	PULA													32	3	1	SP + 1 → SP, Msp → A	•	•	•	•	•	•
	PULB													33	3	1	SP + 1 → SP, Msp → B	•	•	•	•	•	•
Rotate Left	ROL							69	6	2	79	6	3					•	•	↕	↕	⑥	↕
	ROLA													49	1	1		•	•	↕	↕	⑥	↕
	ROLB													59	1	1		•	•	↕	↕	⑥	↕
Rotate Right	ROR							66	6	2	76	6	3					•	•	↕	↕	⑥	↕
	RORA													46	1	1		•	•	↕	↕	⑥	↕
	RORB													56	1	1		•	•	↕	↕	⑥	↕

(Note) Condition Code Register will be explained in Note of Table 17.

(continued)

Table 14 Accumulator, Memory Manipulation Instructions

Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Shift Left Arithmetic	ASL							68	6	2	78	6	3				M, A, B → C b7 b0	•	•	↕	↕	⑥	↕
	ASLA													48	1	1		•	•	↕	↕	⑥	↕
	ASLB													58	1	1		•	•	↕	↕	⑥	↕
Double Shift Left, Arithmetic	ASLD													05	1	1	C A7 ACC A/ ACC B A0 B7 B0	•	•	↕	↕	⑥	↕
Shift Right Arithmetic	ASR							67	6	2	77	6	3				M, A, B b7 b0 C	•	•	↕	↕	⑥	↕
	ASRA													47	1	1		•	•	↕	↕	⑥	↕
	ASRB													57	1	1		•	•	↕	↕	⑥	↕
Shift Right Logical	LSR							64	6	2	74	6	3				M, A, B 0 → b7 b0 C	•	•	R	↕	⑥	↕
	LSRA													44	1	1		•	•	R	↕	⑥	↕
	LSRB													54	1	1		•	•	R	↕	⑥	↕
Double Shift Right Logical	LSRD													04	1	1	0 → ACC A/ ACC B A7 A0 B7 B0 C	•	•	R	↕	⑥	↕
Store Accumulator	STAA				97	3	2	A7	4	2	B7	4	3				A → M	•	•	↕	↕	R	•
	STAB				D7	3	2	E7	4	2	F7	4	3				B → M	•	•	↕	↕	R	•
Store Double Accumulator	STD				DD	4	2	ED	5	2	FD	5	3				A → M B → M + 1	•	•	↕	↕	R	•
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3				A - M → A	•	•	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3				B - M → B	•	•	↕	↕	↕	↕
Double Subtract	SUBD	83	3	3	93	4	2	A3	5	2	B3	5	3				A : B - M : M + 1 → A : B	•	•	↕	↕	↕	↕
Subtract Accumulators	SBA													10	1	1	A - B → A	•	•	↕	↕	↕	↕
Subtract With Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3				A - M - C → A	•	•	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3				B - M - C → B	•	•	↕	↕	↕	↕
Transfer Accumulators	TAB													16	1	1	A → B	•	•	↕	↕	R	•
	TBA													17	1	1	B → A	•	•	↕	↕	R	•
Test Zero or Minus	TST							6D	4	2	7D	4	3				M - 00	•	•	↕	↕	R	R
	TSTA													4D	1	1	A - 00	•	•	↕	↕	R	R
	TSTB													5D	1	1	B - 00	•	•	↕	↕	R	R
And Immediate	AIM				71	6	3	61	7	3							M·IMM→M	•	•	↕	↕	R	•
OR Immediate	OIM				72	6	3	62	7	3							M+IMM→M	•	•	↕	↕	R	•
EOR Immediate	EIM				75	6	3	65	7	3							M⊕IMM→M	•	•	↕	↕	R	•
Test Immediate	TIM				7B	4	3	6B	5	3							M·IMM	•	•	↕	↕	R	•

(Note) Condition Code Register will be explained in Note of Table 17.

- **Additional Instruction**

In addition to the HD6801 instruction set, the HD63701X0 prepares the following new instructions.

AIM..... (M)·(IMM) → (M)

Executes "AND" operation to immediate data and the memory contents and stores its result in the memory.

OIM..... (M) + (IMM) → (M)

Executes "OR" operation to immediate data and the memory contents and stores its result in the memory.

EIM..... (M) ⊕ (IMM) → (M)

Executes "EOR" operation to immediate data and the memory contents and stores its result in the memory.

TIM..... (M) · (IMM)

Executes "AND" operation to immediate data and changes the relative flag of the condition code register.

These area 3-byte instructions; the first byte is op code, the second immediate data and the third address modifier.

XGDX..... (ACCD) ↔ (IX)

Exchanges the contents of accumulator and the index register.

SLP

Goes to the sleep mode. Refer to "LOW POWER DISSIPATION MODE" for more details of the sleep mode.

Table 15 Index Register, Stack Manipulation Instructions

Pointer Operations	Mnemonic	Addressing Modes														Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Compare Index Reg	CPX	8C	3	3	9C	4	2	AC	5	2	BC	5	3				X−M:M+1	•	•	↕	↕	↕	↕
Decrement Index Reg	DEX													09	1	1	X − 1 → X	•	•	•	↕	•	•
Decrement Stack Pntr	DES													34	1	1	SP − 1 → SP	•	•	•	•	•	•
Increment Index Reg	INX													08	1	1	X + 1 → X	•	•	•	↕	•	•
Increment Stack Pntr	INS													31	1	1	SP + 1 → SP	•	•	•	•	•	•
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	5	2	FE	5	3				M → X_H, (M + 1) → X_L	•	•	7	↕	R	•
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	5	2	BE	5	3				M → SP_H, (M+1) → SP_L	•	•	$\overline{7}$	↕	R	•
Store Index Reg	STX				DF	4	2	EF	5	2	FF	5	3				X_H → M, X_L → (M + 1)	•	•	7	↕	R	•
Store Stack Pntr	STS				9F	4	2	AF	5	2	BF	5	3				SP_H → M, SP_L → (M + 1)	•	•	7	↕	R	•
Index Reg → Stack Pntr	TXS													35	1	1	X − 1 → SP	•	•	•	•	•	•
Stack Pntr → Index Reg	TSX													30	1	1	SP + 1 → X	•	•	•	•	•	•
Add	ABX													3A	1	1	B + X → X	•	•	•	•	•	•
Push Data	PSHX													3C	5	1	X_L → M_{sp}, SP − 1 → SP X_H → M_{sp}, SP − 1 → SP	•	•	•	•	•	•
Pull Data	PULX													38	4	1	SP + 1 → SP, M_{sp} → X_H SP + 1 → SP, M_{sp} → X_L	•	•	•	•	•	•
Exchange	X G D X													18	2	1	ACCD··IX	•	•	•	•	•	•

(Note) Condition Code Register will be explained in Note of Table 17.

Table 16 Jump, Branch Instruction

Operations	Mnemonic	Addressing Modes															Branch Test	Condition Code Register					
		RELATIVE			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Branch Always	BRA	20	3	2													None	•	•	•	•	•	•
Branch Never	BRN	21	3	2													None	•	•	•	•	•	•
Branch If Carry Clear	BCC	24	3	2													C = 0	•	•	•	•	•	•
Branch If Carry Set	BCS	25	3	2													C = 1	•	•	•	•	•	•
Branch If = Zero	BEQ	27	3	2													Z = 1	•	•	•	•	•	•
Branch If ≥ Zero	BGE	2C	3	2													N ⊕ V = 0	•	•	•	•	•	•
Branch If > Zero	BGT	2E	3	2													Z + (N ⊕ V) = 0	•	•	•	•	•	•
Branch If Higher	BHI	22	3	2													C + Z = 0	•	•	•	•	•	•
Branch If ≤ Zero	BLE	2F	3	2													Z + (N ⊕ V) = 1	•	•	•	•	•	•
Branch If Lower Or Same	BLS	23	3	2													C + Z = 1	•	•	•	•	•	•
Branch If < Zero	BLT	2D	3	2													N ⊕ V = 1	•	•	•	•	•	•
Branch If Minus	BMI	2B	3	2													N = 1	•	•	•	•	•	•
Branch If Not Equal Zero	BNE	26	3	2													Z = 0	•	•	•	•	•	•
Branch If Overflow Clear	BVC	28	3	2													V = 0	•	•	•	•	•	•
Branch If Overflow Set	BVS	29	3	2													V = 1	•	•	•	•	•	•
Branch If Plus	BPL	2A	3	2													N = 0	•	•	•	•	•	•
Branch To Subroutine	BSR	8D	5	2														•	•	•	•	•	•
Jump	JMP							6E	3	2	7E	3	3					•	•	•	•	•	•
Jump To Subroutine	JSR				9D	5	2	AD	5	2	BD	6	3					•	•	•	•	•	•
No Operation	NOP													01	1	1	Advances Prog. Cntr. Only	•	•	•	•	•	•
Return From Interrupt	RTI													3B	10	1		——— ⑧ ———					
Return From Subroutine	RTS													39	5	1		•	•	•	•	•	•
Software Interrupt	SWI													3F	12	1		•	S	•	•	•	•
Wait for Interrupt*	WAI													3E	9	1		•	⑨	•	•	•	•
Sleep	SLP													1A	4	1		•	•	•	•	•	•

(Note) * WAI puts R/$\overline{W}$ high; Address Bus goes to FFFF; Data Bus goes to the three state.
Condition Code Register will be explained in Note of Table 17.

Table 17 Condition Code Register Manipulation Instructions

Operations	Mnemonic	Addressing Modes IMPLIED OP	~	#	Boolean Operation	Condition Code Register 5 H	4 I	3 N	2 Z	1 V	0 C
Clear Carry	CLC	0C	1	1	0 → C	•	•	•	•	•	R
Clear Interrupt Mask	CLI	0E	1	1	0 → I	•	R	•	•	•	•
Clear Overflow	CLV	0A	1	1	0 → V	•	•	•	•	R	•
Set Carry	SEC	0D	1	1	1 → C	•	•	•	•	•	S
Set Interrupt Mask	SEI	0F	1	1	1 → I	•	S	•	•	•	•
Set Overflow	SEV	0B	1	1	1 → V	•	•	•	•	S	•
Accumulator A → CCR	TAP	06	1	1	A → CCR	——— ⑩ ———					
CCR → Accumulator A	TPA	07	1	1	CCR → A	•	•	•	•	•	•

LEGEND

OP Operation Code (Hexadecimal)
~ Number of MCU Cycles
M_{SP} Contents of memory location pointed to by Stack Pointer
Number of Program Bytes
+ Arithmetic Plus
– Arithmetic Minus
• Boolean AND
+ Boolean Inclusive OR
⊕ Boolean Exclusive OR
$\overline{M}$ Complement of M
→ Transfer into
0 Bit = Zero
00 Byte = Zero

CONDITION CODE SYMBOLS

H Half-carry from bit 3 to bit 4
I Interrupt mask
N Negative (sign bit)
Z Zero (byte)
V Overflow, 2's complement
C Carry/Borrow from/to bit 7
R Reset Always
S Set Always
↕ Set if true after test or clear
• Not Affected

(Note) Condition Code Register Notes: (Bit set if test is true and cleared otherwise)

① (Bit V) Test: Result = 10000000?
② (Bit C) Test: Result ≠ 00000000?
③ (Bit C) Test: BCD Character of high-order byte greater than 10? (Not cleared if previously set)
④ (Bit V) Test: Operand = 10000000 prior to execution?
⑤ (Bit V) Test: Operand = 01111111 prior to execution?
⑥ (Bit V) Test: Set equal to N⊕ C = 1 after the execution of instructions
⑦ (Bit N) Test: Result less than zero? (Bit 15=1)
⑧ (All Bit) Load Condition Code Register from Stack.
⑨ (Bit I) Set when interrupt occurs. If previously set, a Non-Maskable Interrupt is required to exist the wait state.
⑩ (All Bit) Set according to the contents of Accumulator A.
⑪ (Bit C) Result of Multiplication Bit 7=1? (ACCB)

Table 18 OP-Code Map

OP CODE HI / LO		0000 0	0001 1	0010 2	0011 3	ACC A 0100 4	ACC B 0101 5	IND 0110 6	EXT / DIR* 0111 7	ACCA or SP IMM 1000 8	DIR 1001 9	IND 1010 A	EXT 1011 B	ACCB or X IMM 1100 C	DIR 1101 D	IND 1110 E	EXT 1111 F	
0000	0		SBA	BRA	TSX	NEG				SUB								0
0001	1	NOP	CBA	BRN	INS			AIM		CMP								1
0010	2			BHI	PULA			OIM		SBC								2
0011	3			BLS	PULB	COM				SUBD				ADDD				3
0100	4	LSRD		BCC	DES	LSR				AND								4
0101	5	ASLD		BCS	TXS			EIM		BIT								5
0110	6	TAP	TAB	BNE	PSHA	ROR				LDA								6
0111	7	TPA	TBA	BEQ	PSHB	ASR					STA				STA			7
1000	8	INX	XGDX	BVC	PULX	ASL				EOR								8
1001	9	DEX	DAA	BVS	RTS	ROL				ADC								9
1010	A	CLV	SLP	BPL	ABX	DEC				ORA								A
1011	B	SEV	ABA	BMI	RTI			TIM		ADD								B
1100	C	CLC		BGE	PSHX	INC				CPX				LDD				C
1101	D	SEC		BLT	MUL	TST				BSR	JSR				STD			D
1110	E	CLI		BGT	WAI			JMP		LDS				LDX				E
1111	F	SEI		BLE	SWI	CLR					STS				STX			F
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

UNDEFINED OP CODE

* Only each instructions of AIM, OIM, EIM, TIM

■ CPU OPERATION

• CPU Instruction Flow

When operating, the CPU fetches an instruction from a memory and executes the required functions. This sequence starts after the reset release and repeats itself limitlessly if not affected by a special instruction or a control signal. SWI, RTI, WAI and SLP instructions are to change this operation, while $\overline{NMI}$, $\overline{IRQ_1}$, $\overline{IRQ_2}$, IRQ_3, $\overline{HALT}$ and $\overline{STBY}$ are to control it. Fig. 28 gives the CPU mode shift and Fig. 29 the CPU system flowchart. Table 19 shows the CPU operating states and port states.

• Operation at Each Instruction Cycle

Table 20 provides the operation at each instruction cycle. By the pipeline control of the HD63701X0, MULT, PUL, DAA and XGDX instructions etc. prefetch the next instruction. So attention is necessary to the counting of the instruction cycles because it is different from the existent one ----- op code fetch to the next instruction op code.

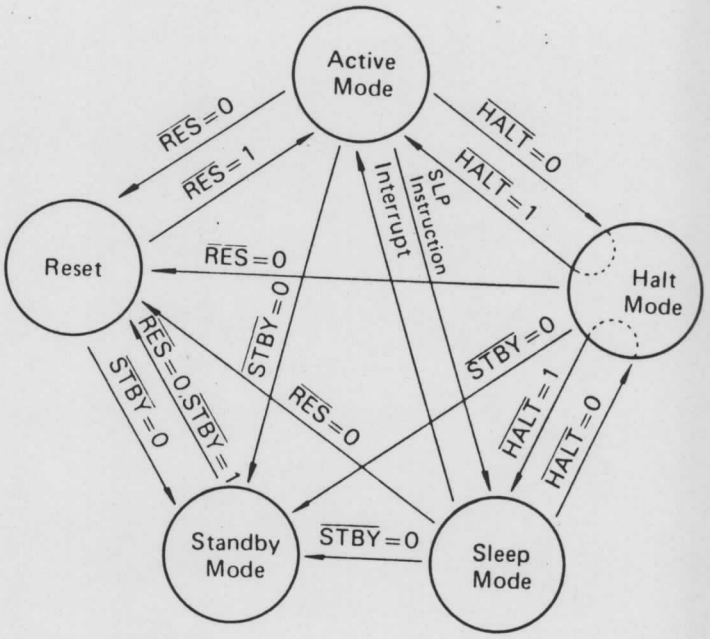


Figure 28 CPU Operation Mode Transition

Table 19 CPU Operation State and Port State

Port	Mode	Reset	STBY****	HALT***	Sleep
Port 1 ($A_0 \sim A_7$)	Mode 1, 2	H	T	T	H
	Mode 3	T			Keep
Port 2	Mode 1, 2	T	T	Keep	Keep
	Mode 3				
Port 3 ($D_0 \sim D_7$)	Mode 1, 2	T	T	T	T
	Mode 3				Keep
Port 4 ($A_8 \sim A_{15}$)	Mode 1, 2	H	T	T	H
	Mode 3	T			Keep
Port 5	Mode 1, 2	T	T	T	T
	Mode 3				
Port 6	Mode 1, 2	T	T	Keep	Keep
	Mode 3				
Port 7	Mode 1, 2	*	T	**	*
	Mode 3	T			Keep

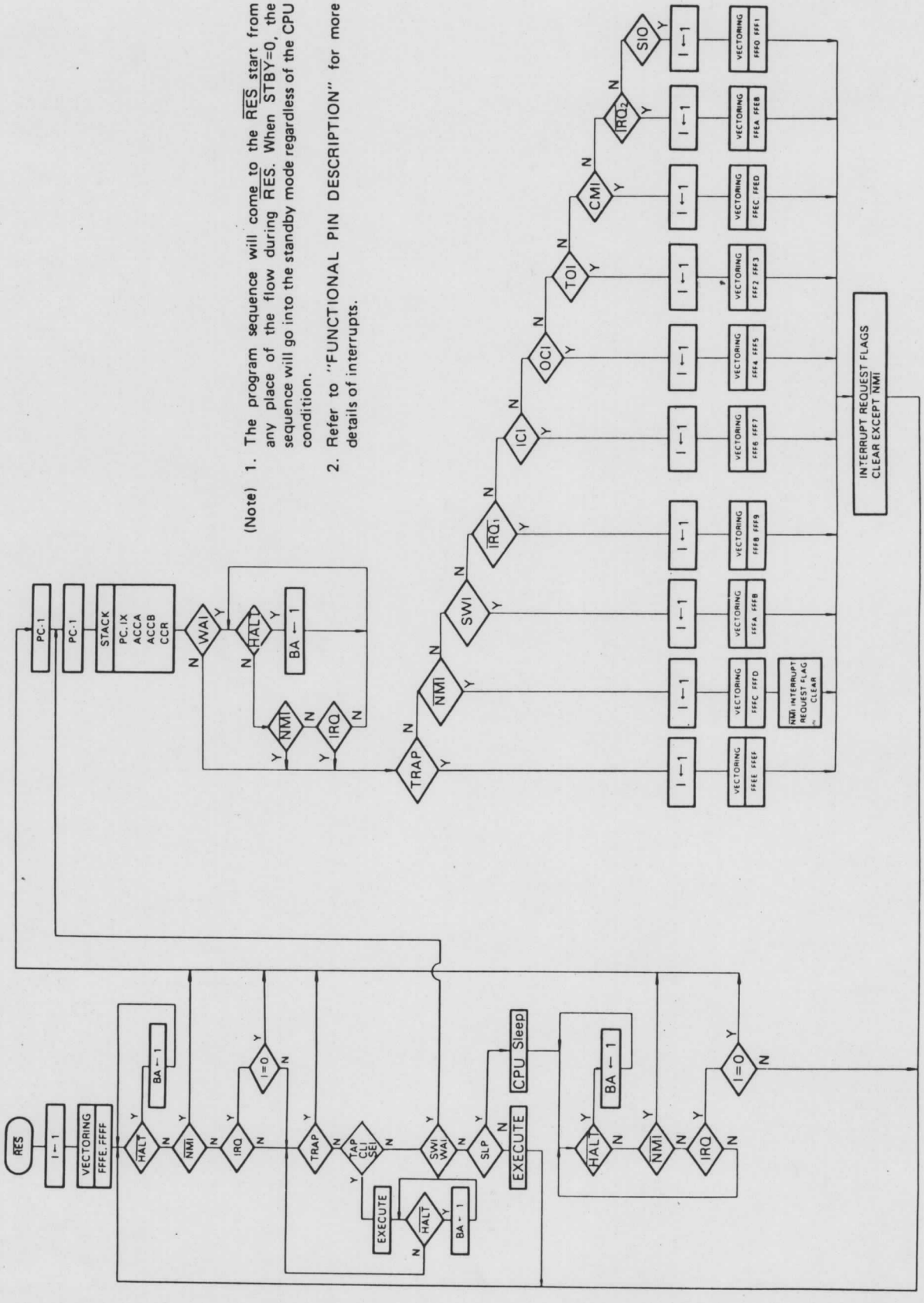
H ; High, L ; Low, T ; High Impedance

* $\overline{RD}$, $\overline{WR}$, R/$\overline{W}$, $\overline{LIR}$=H, BA=L

** $\overline{RD}$, $\overline{WR}$, R/$\overline{W}$=T, $\overline{LIR}$, BA=H

*** $\overline{HALT}$ is unacceptable in mode 3.

**** E pin goes to high impedance state.



(Note) 1. The program sequence will come to the $\overline{RES}$ start from any place of the flow during $\overline{RES}$. When $\overline{STBY}$=0, the sequence will go into the standby mode regardless of the CPU condition.

2. Refer to "FUNCTIONAL PIN DESCRIPTION" for more details of interrupts.

Figure 29 HD63701X0 System Flow Chart

Table 20 Cycle-by-Cycle Operation

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
IMMEDIATE								
ADC ADD AND BIT CMP EOR LDA ORA SBC SUB	2	1	Op Code Address + 1	1	0	1	1	Operand Data
		2	Op Code Address + 2	1	0	1	0	Next Op Code
ADDD CPX LDD LDS LDX SUBD	3	1	Op Code Address + 1	1	0	1	1	Operand Data (MSB)
		2	Op Code Address + 2	1	0	1	1	Operand Data (LSB)
		3	Op Code Address + 3	1	0	1	0	Next Op Code
DIRECT								
ADC ADD AND BIT CMP EOR LDA ORA SBC SUB	3	1	Op Code Address + 1	1	0	1	1	Address of Operand (LSB)
		2	Address of Operand	1	0	1	1	Operand Data
		3	Op Code Address + 2	1	0	1	0	Next Op Code
STA	3	1	Op Code Address + 1	1	0	1	1	Destination Address
		2	Destination Address	0	1	0	1	Accumulator Data
		3	Op Code Address + 2	1	0	1	0	Next Op Code
ADDD CPX LDD LDS LDX SUBD	4	1	Op Code Address + 1	1	0	1	1	Address of Operand (LSB)
		2	Address of Operand	1	0	1	1	Operand Data (MSB)
		3	Address of Operand + 1	1	0	1	1	Operand Data (LSB)
		4	Op Code Address + 2	1	0	1	0	Next Op Code
STD STS STX	4	1	Op Code Address + 1	1	0	1	1	Destination Address (LSB)
		2	Destination Address	0	1	0	1	Register Data (MSB)
		3	Destination Address + 1	0	1	0	1	Register Data (LSB)
		4	Op Code Address + 2	1	0	1	0	Next Op Code
JSR	5	1	Op Code Address + 1	1	0	1	1	Jump Address (LSB)
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer - 1	0	1	0	1	Return Address (MSB)
		5	Jump Address	1	0	1	0	First Subroutine Op Code
TIM	4	1	Op Code Address + 1	1	0	1	1	Immediate Data
		2	Op Code Address + 2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data
		4	Op Code Address + 3	1	0	1	0	Next Op Code
AIM EIM OIM	6	1	Op Code Address + 1	1	0	1	1	Immediate Data
		2	Op Code Address + 2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data
		4	FFFF	1	1	1	1	Restart Address (LSB)
		5	Address of Operand	0	1	0	1	New Operand Data
		6	Op Code Address + 3	1	0	1	0	Next Op Code

(Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
INDEXED								
JMP	3	1	Op Code Address + 1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Jump Address	1	0	1	0	First Op Code of Jump Routine
ADC ADD AND BIT CMP EOR LDA ORA SBC SUB TST	4	1	Op Code Address + 1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX + Offset	1	0	1	1	Operand Data
		4	Op Code Address + 2	1	0	1	0	Next Op Code
STA	4	1	Op Code Address + 1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX + Offset	0	1	0	1	Accumulator Data
		4	Op Code Address + 2	1	0	1	0	Next Op Code
ADDD CPX LDD LDS LDX SUBD	5	1	Op Code Address + 1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX + Offset	1	0	1	1	Operand Data (MSB)
		4	IX + Offset + 1	1	0	1	1	Operand Data (LSB)
		5	Op Code Address + 2	1	0	1	0	Next Op Code
STD STS STX	5	1	Op Code Address + 1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX + Offset	0	1	0	1	Register Data (MSB)
		4	IX + Offset + 1	0	1	0	1	Register Data (LSB)
		5	Op Code Address + 2	1	0	1	0	Next Op Code
JSR	5	1	Op Code Address + 1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer − 1	0	1	0	1	Return Address (MSB)
		5	IX + Offset	1	0	1	0	First Subroutine Op Code
ASL ASR COM DEC INC LSR NEG ROL ROR	6	1	Op Code Address + 1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX + Offset	1	0	1	1	Operand Data
		4	FFFF	1	1	1	1	Restart Address (LSB)
		5	IX + Offset	0	1	0	1	New Operand Data
		6	Op Code Address + 1	1	0	1	0	Next Op Code
TIM	5	1	Op Code Address + 1	1	0	1	1	Immediate Data
		2	Op Code Address + 2	1	0	1	1	Offset
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	IX + Offset	1	0	1	1	Operand Data
		5	Op Code Address + 3	1	0	1	0	Next Op Code
CLR	5	1	Op Code Address + 1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX + Offset	1	0	1	1	Operand Data
		4	IX + Offset	0	1	0	1	00
		5	Op Code Address + 2	1	0	1	0	Next Op Code
AIM EIM OIM	7	1	Op Code Address + 1	1	0	1	1	Immediate Data
		2	Op Code Address + 2	1	0	1	1	Offset
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	IX + Offset	1	0	1	1	Operand Data
		5	FFFF	1	1	1	1	Restart Address (LSB)
		6	IX + Offset	0	1	0	1	New Operand Data
		7	Op Code Address + 3	1	0	1	0	Next Op Code

(Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
EXTEND								
JMP	3	1	Op Code Address + 1	1	0	1	1	Jump Address (MSB)
		2	Op Code Address + 2	1	0	1	1	Jump Address (LSB)
		3	Jump Address	1	0	1	0	Next Op Code
ADC ADD TST AND BIT CMP EOR LDA ORA SBC SUB	4	1	Op Code Address + 1	1	0	1	1	Address of Operand (MSB)
		2	Op Code Address + 2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data
		4	Op Code Address + 3	1	0	1	0	Next Op Code
STA	4	1	Op Code Address + 1	1	0	1	1	Destination Address (MSB)
		2	Op Code Address + 2	1	0	1	1	Destination Address (LSB)
		3	Destination Address	0	1	0	1	Accumulator Data
		4	Op Code Address + 3	1	0	1	0	Next Op Code
ADDD CPX LDD LDS LDX SUBD	5	1	Op Code Address + 1	1	0	1	1	Address of Operand (MSB)
		2	Op Code Address + 2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data (MSB)
		4	Address of Operand + 1	1	0	1	1	Operand Data (LSB)
		5	Op Code Address + 3	1	0	1	0	Next Op Code
STD STS STX	5	1	Op Code Address + 1	1	0	1	1	Destination Address (MSB)
		2	Op Code Address + 2	1	0	1	1	Destination Address (LSB)
		3	Destination Address	0	1	0	1	Register Data (MSB)
		4	Destination Address + 1	0	1	0	1	Register Data (LSB)
		5	Op Code Address + 3	1	0	1	0	Next Op Code
JSR	6	1	Op Code Address + 1	1	0	1	1	Jump Address (MSB)
		2	Op Code Address + 2	1	0	1	1	Jump Address (LSB)
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	Stack Pointer	0	1	0	1	Return Address (LSB)
		5	Stack Pointer − 1	0	1	0	1	Return Address (MSB)
		6	Jump Address	1	0	1	0	First Subroutine Op Code
ASL ASR COM DEC INC LSR NEG ROL ROR	6	1	Op Code Address + 1	1	0	1	1	Address of Operand (MSB)
		2	Op Code Address + 2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data
		4	FFFF	1	1	1	1	Restart Address (LSB)
		5	Address of Operand	0	1	0	1	New Operand Data
		6	Op Code Address + 3	1	0	1	0	Next Op Code
CLR	5	1	Op Code Address + 1	1	0	1	1	Address of Operand (MSB)
		2	Op Code Address + 2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data
		4	Address of Operand	0	1	0	1	00
		5	Op Code Address + 3	1	0	1	0	Next Op Code

(Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
IMPLIED								
ABA ABX ASL ASLD ASR CBA CLC CLI CLR CLV COM DEC DES DEX INC INS INX LSR LSRD ROL ROR NOP SBA SEC SEI SEV TAB TAP TBA TPA TST TSX TXS	1	1	Op Code Address + 1	1	0	1	0	Next Op Code
DAA XGDX	2	1	Op Code Address + 1	1	0	1	0	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
PULA PULB	3	1	Op Code Address + 1	1	0	1	0	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer + 1	1	0	1	1	Data from Stack
PSHA PSHB	4	1	Op Code Address + 1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Accumulator Data
		4	Op Code Address + 1	1	0	1	0	Next Op Code
PULX	4	1	Op Code Address + 1	1	0	1	0	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer + 1	1	0	1	1	Data from Stack (MSB)
		4	Stack Pointer + 2	1	0	1	1	Data from Stack (LSB)
PSHX	5	1	Op Code Address + 1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Index Register (LSB)
		4	Stack Pointer − 1	0	1	0	1	Index Register (MSB)
		5	Op Code Address + 1	1	0	1	0	Next Op Code
RTS	5	1	Op Code Address + 1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer + 1	1	0	1	1	Return Address (MSB)
		4	Stack Pointer + 2	1	0	1	1	Return Address (LSB)
		5	Return Address	1	0	1	0	First Op Code of Return Routine
MUL	7	1	Op Code Address + 1	1	0	1	0	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	FFFF	1	1	1	1	Restart Address (LSB)
		5	FFFF	1	1	1	1	Restart Address (LSB)
		6	FFFF	1	1	1	1	Restart Address (LSB)
		7	FFFF	1	1	1	1	Restart Address (LSB)

(Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
IMPLIED								
WAI	9	1	Op Code Address + 1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer − 1	0	1	0	1	Return Address (MSB)
		5	Stack Pointer − 2	0	1	0	1	Index Register (LSB)
		6	Stack Pointer − 3	0	1	0	1	Index Register (MSB)
		7	Stack Pointer − 4	0	1	0	1	Accumulator A
		8	Stack Pointer − 5	0	1	0	1	Accumulator B
		9	Stack Pointer − 6	0	1	0	1	Conditional Code Register
RTI	10	1	Op Code Address + 1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer + 1	1	0	1	1	Conditional Code Register
		4	Stack Pointer + 2	1	0	1	1	Accumulator B
		5	Stack Pointer + 3	1	0	1	1	Accumulator A
		6	Stack Pointer + 4	1	0	1	1	Index Register (MSB)
		7	Stack Pointer + 5	1	0	1	1	Index Register (LSB)
		8	Stack Pointer + 6	1	0	1	1	Return Address (MSB)
		9	Stack Pointer + 7	1	0	1	1	Return Address (LSB)
		10	Return Address	1	0	1	0	First Op Code of Return Routine
SWI	12	1	Op Code Address + 1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer − 1	0	1	0	1	Return Address (MSB)
		5	Stack Pointer − 2	0	1	0	1	Index Register (LSB)
		6	Stack Pointer − 3	0	1	0	1	Index Register (MSB)
		7	Stack Pointer − 4	0	1	0	1	Accumulator A
		8	Stack Pointer − 5	0	1	0	1	Accumulator B
		9	Stack Pointer − 6	0	1	0	1	Conditional Code Register
		10	Vector Address FFFA	1	0	1	1	Address of SWI Routine (MSB)
		11	Vector Address FFFP	1	0	1	1	Address of SWI Routine (LSB)
		12	Address of SWI Routine	1	0	1	0	First Op Code of SWI Routine
SLP	4	1	Op Code Address + 1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		Sleep						
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	Op Code Address + 1	1	0	1	0	Next Op Code
RELATIVE								
BCC BCS BEQ BGE BGT BHI BLE BLS BLT BMT BNE BPL BRA BRN BVC BVS	3	1	Op Code Address + 1	1	0	1	1	Branch Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Branch Address ····Test = "1" Op Code Address + 1 ···Test = "0"	1	0	1	0	First Op Code of Branch Routine Next Op Code
BSR	5	1	Op Code Address + 1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer − 1	0	1	0	1	Return Address (MSB)
		5	Branch Address	1	0	1	0	First Op Code of Subroutine

■ APPLICATION NOTES

● The PROM Programming and Maintenance

(1) The PROM Programming and Data Retention

A ZTAT™ MICRO's memory cell is the same as an EPROM device and it is programmed by hot electrons injected to the floating gate with applying high voltage at the control gate and the drain. The electrons have been trapped by the potential barrier at the polysilicon-oxide (SiO_2) by which the floating gate is completely sorrounded. The programmed cell becomes a "0".

The memory cell will be discharged by;

① Heat; discharged by thermal emitting electrons

② Applied with high voltage; discharged by high electric field.

Charge loss from the normal cell is negligible. But if there are some defects at the SiO_2, the cell will be rapidly discharged through the defects by heat or high voltage. Such a defective part is rejected by manufacturing screenings.

The erased, or discharged, cell is a "1".

(2) Precaution of the PROM Programming

The PROM memory cell should be programmed with the specified voltage and timing. The higher program voltage V_{PP} or the longer program pulse width t_{PW} is applied, the more quantity of electrons will be injected to the floating gate. However, a p-n junction will be broken permanently if V_{PP} is applied to more than maximum ratings. Especially V_{PP} overshoot of a PROM programmer should be checked.

Negative-noise to device pins may cause a parasitic transistor effect and reduce the breakdown voltage.

(3) Screening procedure of the ZTAT™ MICRO

In general, any standard manufacturing screening of semiconductor devices will make initial failures rejected and improve reliability. The bake procedure is the standard screening for EPROM devices, which accelerates any electron leakage at the floating gate described before (see (1)). The manufacturer, of course, tests the CPU, RAM, I/O, other logic functions and the PROM of the ZTAT™ MICRO after screening at wafer site, and rejects any devices which do not pass the tests. If users require improving reliability of the ZTAT™ MICRO's PROM portion, it is recommended that users also carry out the screening procedure shown in Fig. 31 after programming.

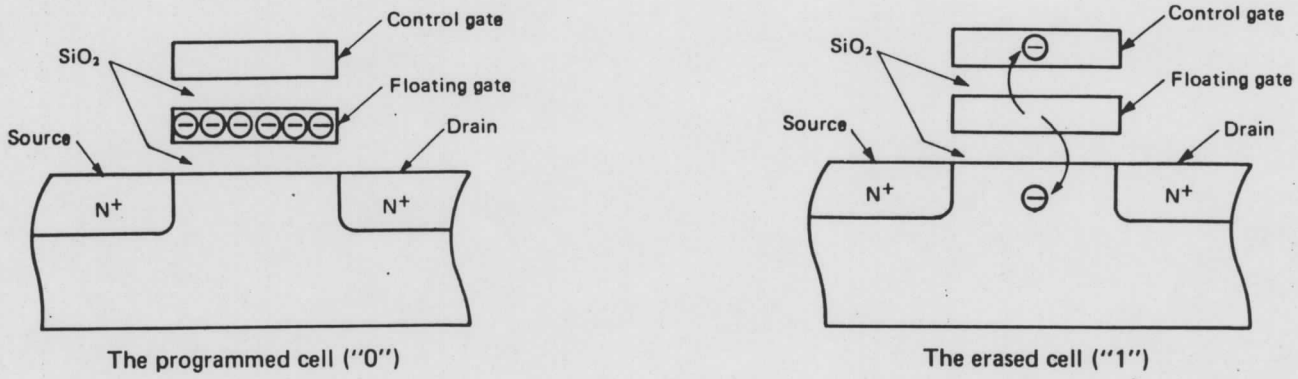


Figure 30 Cross-section of An EPROM Memory Cell

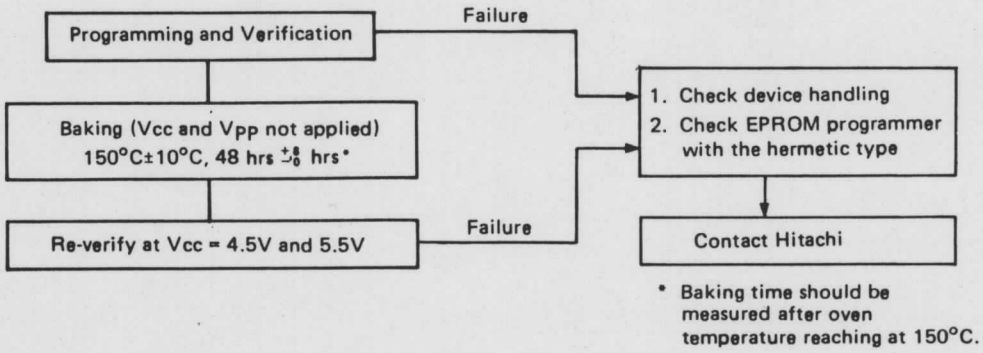


Figure 31 Recommended Screening Procedure of the ZTAT™ MICRO

(Caution) If the user experiences several consecutive programming failures, from same EPROM programmer, after the recommended screening procedure shown in Fig. 31, then call Hitachi.

(4) EPROM programmers and socket adapters

EPROM programmers and socket adapters which are recommended for the HD63701X0 are shown Table 21.

A socket adapter is a tool to convert from 64-pin socket to standard 24-pin socket.

Table 21 EPROM Programmers and Socket Adapters for the HD63701X0

EPROM Programmer		Socket Adapter	
Maker	Type No.	Maker	Type No.
DATA I/O (U.S.A.)	12A/121B 22A/22B 29A/29B	Hitachi Ltd.	H67PWA01A
		Data I/O	HD63701X0 (for 29A/29B)
AVAL CORP. (JAPAN)	PKW-1000 PKW-7000	Hitachi Ltd.	H67PWA01B
Minato Electronics Inc. (JAPAN)	M1863 M1866 7GU-2700	Hitachi Ltd.	H67PWA01B

- **Write only Register**

When a write-only register such as the DDR of the port is read by the MPU, "$FF" always appears on the data bus. Note that when an instruction which reads the memory contents and does some arithmetic operation on the contents of the write-only register, it always gets $FF as the arithmetic and logical results. AIM, OIM and EIM instructions are unable to apply especially for the bit manipulation of the DDR of the I/O port.

- **Trap Interrupt**

When execution an RTI instruction at the end of the interrupt routine, trap interrupt different from other interrupts returns to the address where the trap interrupt was generated. Attention is necessary when using several trap interrupts in the program. See Fig. 32 and 33 for details.

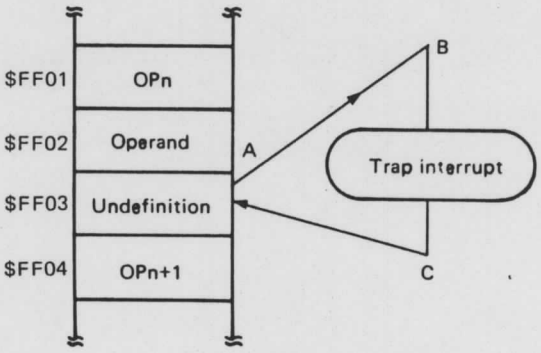


Figure 32 Fetching an Undefined Op-code

After executing OPn instruction, the HD63701X0 fetches and decodes and undefined op-code inside to generate a trap interrupt. When RTI instruction is executed in this trap interrupt servicing routine, the HD63701X0 will set $FF03 in PC, fetch the undefined code again, generate a trap interrupt and repeat ABC endless-loop.

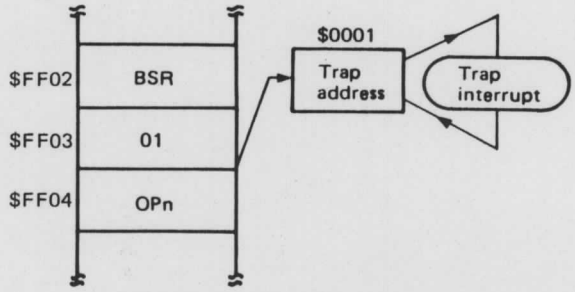


Figure 33 Fetching Erroneously

After performing BSR instruction, the branch destination address is output on an address bus to fetch the first op-code of a subroutine. If $0001 is output as an address by some mistake the HD63701X0 decodes it inside and generates a trap interrupt. When RTI instruction is performed in this trap interrupt servicing routine, the HD63701X0 will set $0001 in PC and start from this address, which causes a trap interrupt again and repeat this endless-loop.

- **Precaution for using WAI instruction**

If $\overline{\text{HALT}}$ turns "Low" in WAI execution, a CPU upset may occur since the correct vector will not be fetched after the halt state has been released. It is recommended to use BRA instruction etc. for software interrupt before $\overline{\text{HALT}}$ turns "Low" shown Fig. 35.

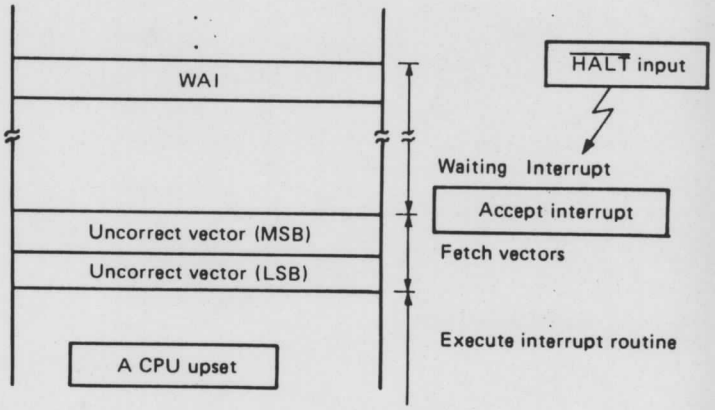


Figure 34 A CPU Upset after $\overline{\text{HALT}}$ Input in WAI Execution

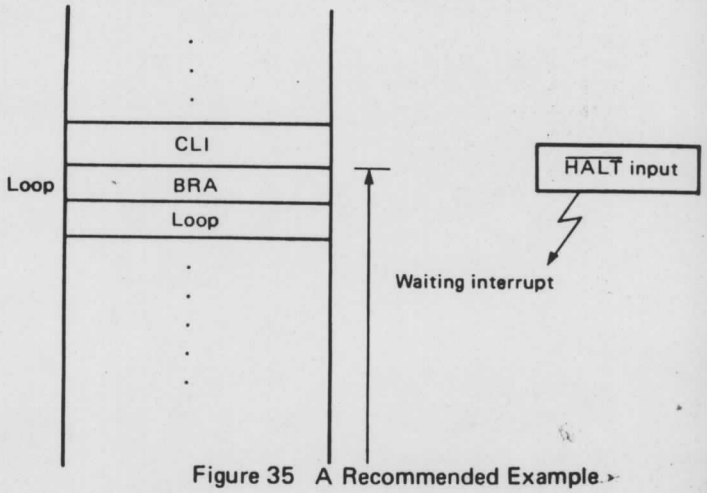


Figure 35 A Recommended Example

● **Power-on Reset**

At power-on it is necessary to hold $\overline{RES}$ "low" to reset the internal state of the device and to provide sufficient time for the oscillator to stabilize. Pay attention to the following.

* Just after power-on, the MPU doesn't enter reset state until the oscillation starts. This is because the reset signal is input internally, with the clocked synchronization as shown below.

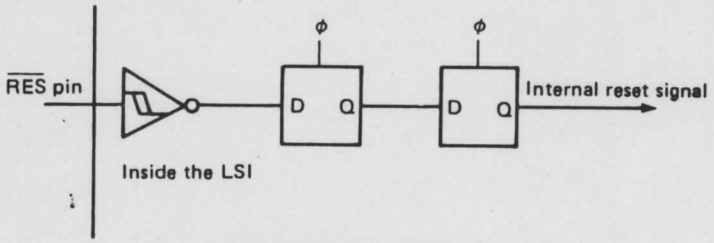


Figure 36 Reset Circuit

Thus, just after power-on the LSI state (I/O port, mode condition etc.) is unstable until the oscillation starts. If it is necessary to inform the LSI state to the external devices during this period, it needs to be done by the external circuits.

● **Board Design of Oscillation Circuits**

Keep the following in mind when connecting a crystal resonator to XTAL and EXTAL pins of the HD63701X0.

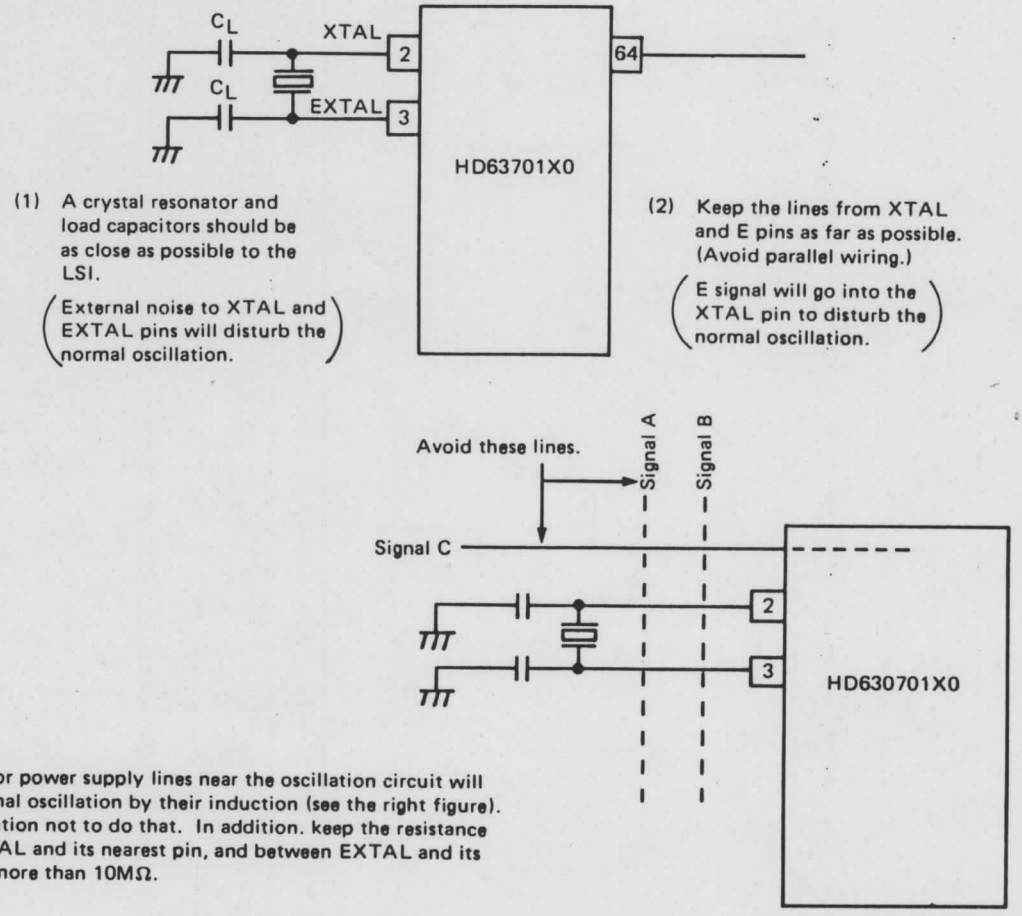


Figure 37 Precaution on Board Design of Oscillation Circuits

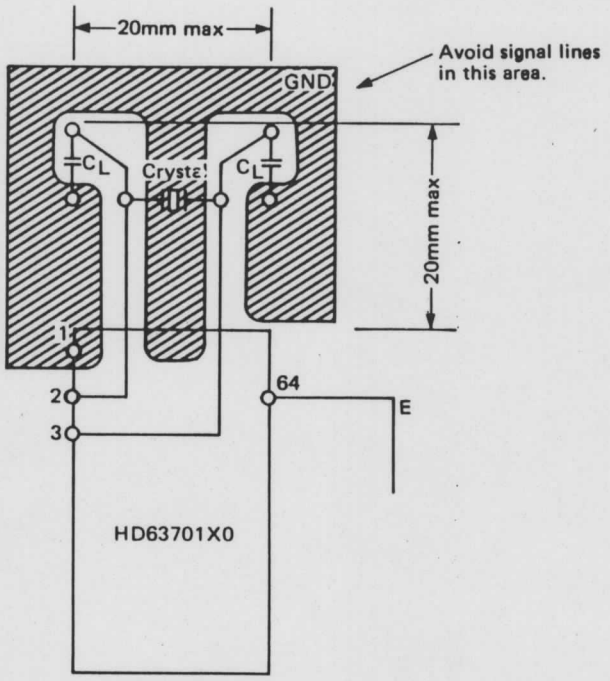


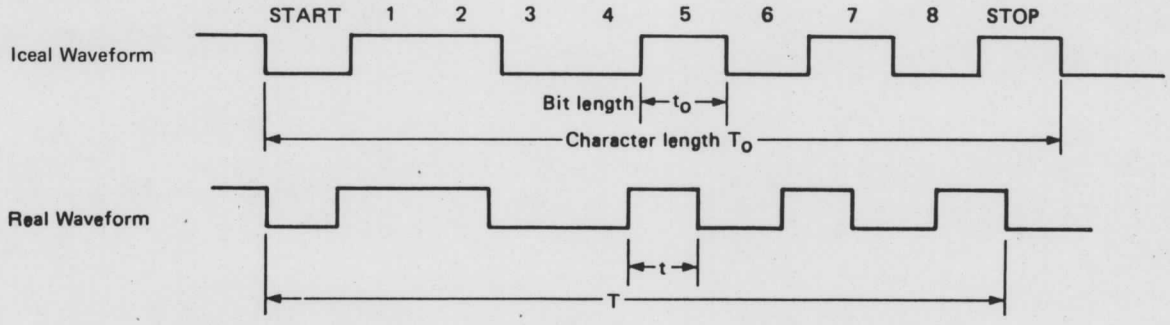
Figure 38 Example of Oscillation Circuits in Board Design

- **Receive Margin of the SCI**

Receive margin of the SCI contained in the HD63701X0 is shown in Table 22.
Note: SCI = Serial Communication Interface.

Table 22

Bit distortion tolerance $(t-t_O) / t_O$	Character distortion tolerance $(T-To) /To$
±43.7%	±4.37%



- **Cautions for OCF clearing in the TCSR of Timer 1**

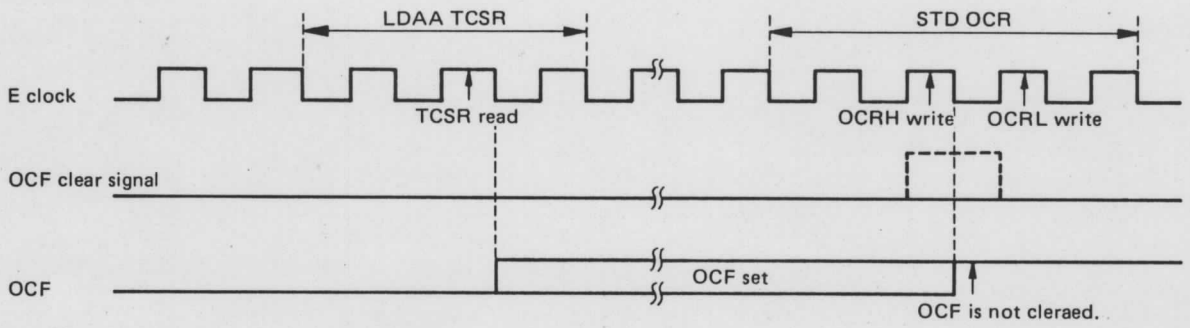
OCF is cleared by writing to the OCR after reading the TCSR at OCF = "1". OCF, however, is not cleared under following conditions;

(1) The TCSR is read at OCF = "0", and then OCF is set (when the comparematch is found between the FCR and the OCR) before writing to the OCR.

(2) The TCSR is read at OCF = "1", and then the compare-match is found between the FCR and the OCR at the OCR write cycle (OCF clear cycle).

1. When OCF is not cleared

(1) case 1



(2) case 2

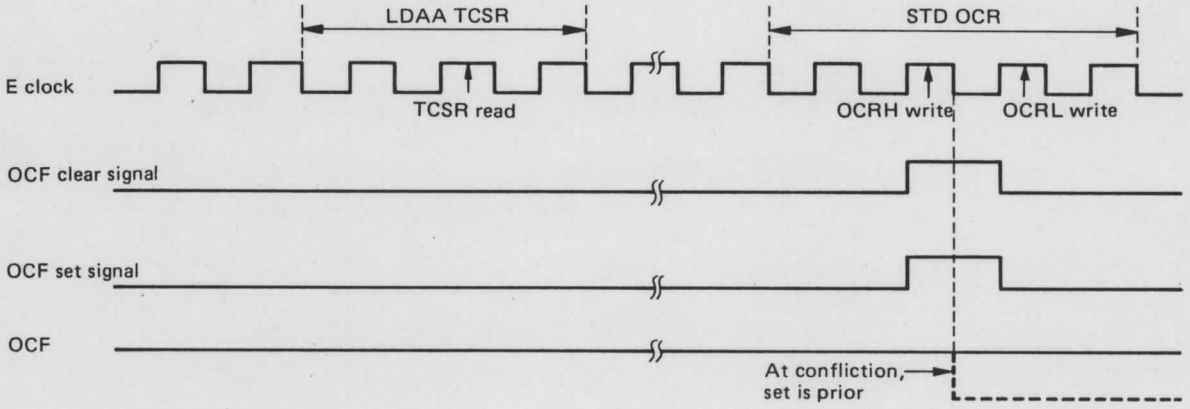


Fig. 39

2. Countermeasures

Following countermeasures against the above cases should be taken to avoid the comparematch during the TCSR read to the OCR write.

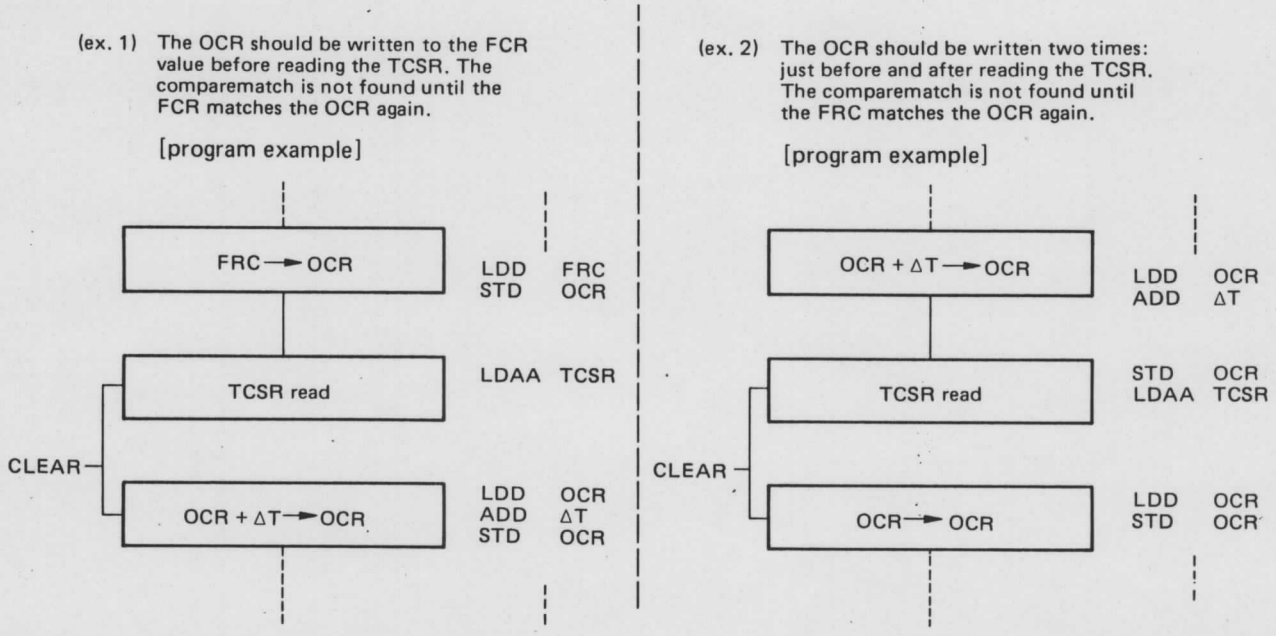


Fig. 40

■ **PACKAGE DIMENSIONS** (Unit: mm)

● **DP-64S**